

- *Developing in Lotus Notes*
- *Application Design Tips*
- *Reusing Objects*

July/August 1995

SPOTLIGHT:

**Lotus
NOTES**

OS/2

Developer

THE MAGAZINE FOR ADVANCED SOFTWARE DEVELOPMENT

A Closer Look at
CORBA

Parts is Parts

**Interflox:
A REXX API**

BUYERS GUIDE:
OOP Tools

Product Review:
**IBM Developer
Connection**

**THE NUTS
AND BOLTS
OF**

**OBJECT-ORIENTED
PROGRAMMING**

U.S. \$9.95 Vol. 7 No. 4



A Miller Freeman Publication

WHEN I WANT ADVICE ABOUT OBJECT-ORIENTED
PROGRAMMING, I'LL ASK AN EXPERT.



SO, ASK US.

You already know reusable class libraries promise a dramatic increase in productivity and efficiency for professional application developers. You also know class libraries created with one C++ compiler can't link with code created by another C++ compiler. And that libraries written in one language can't be used with client code written in another language.

But here's something you might not know: MetaWare and IBM are changing all that.

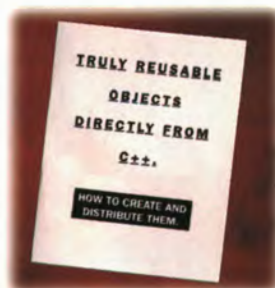
Out With The Old. In With The New. The tight binary coupling that exists between object-oriented class libraries and client code means that changes to either—no matter how small—require a complete recompilation not only of the class itself, but of all the client modules as well.

But with IBM's System Object Model (SOM) and MetaWare's High C/C++ DirectToSOM compiler for OS/2, developers can break that tight binary coupling and extend the advantages of procedure libraries to object-oriented technology.

Pretty amazing, right? And right now, you think this is where our pitch to sell you a compiler comes in. But you're wrong.

Information Only. And It's Free. We know what you really want is information. That way you can make up your own mind about the benefits of SOM and the DirectToSOM High C/C++ compiler for OS/2. We've prepared a white paper that will help. It's called Truly Reusable Objects Directly From C++. How to Create and Distribute Them. It's yours just for the asking.

To receive your free information, call, fax, or e-mail MetaWare at the numbers shown below. Once you review the facts for yourself, you'll understand why MetaWare and IBM mean "objects made easy."



CALL 408 429 6382

FAX 408 429 9273

OR

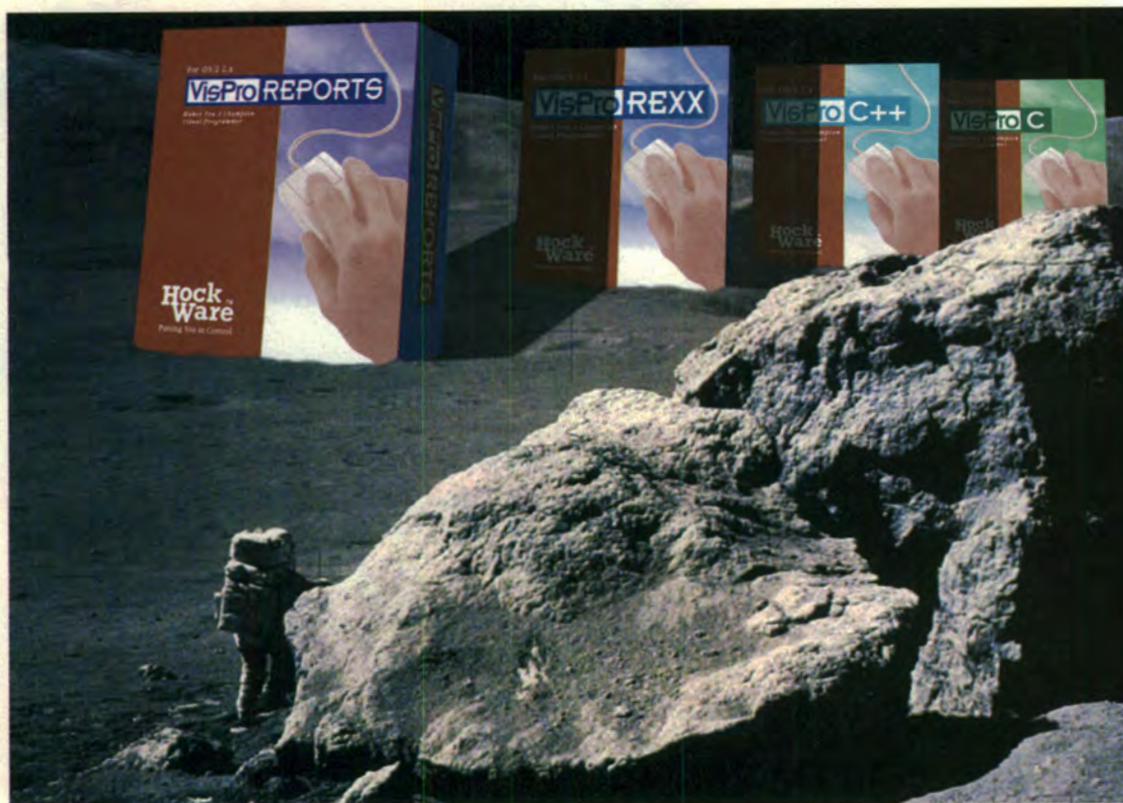
E-MAIL TECHSALES@METAWARE.COM

FOR YOUR FREE COPY.

TODAY.

 **MetaWare®**
OBJECTS MADE EASY.

In the Race for Solutions, Finish First



Introducing **VisPro/Reports**, the first programmable report writer for **VisPro/REXX**, **VX REXX** and **OS/2 REXX**

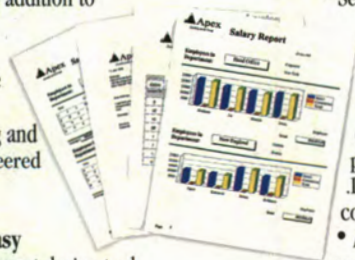
On April 28, 1993 HockWare Inc. shipped VisPro/REXX, the first OS/2 visual programming tool for REXX. Since then, we've achieved many other firsts. VisPro/REXX, VisPro/C and VisPro/C++ are the first REXX, C and C++ tools to incorporate drag and drop programming, an entity-relationship database designer, and a rich set of add-on objects including: spreadsheet, business graphics, formatted entry field, clock and calendar. Whether you're using REXX, C-Presentation Manager, or IBM's User-Interface Class Library, there is a VisPro product to meet your needs.

Yet Another First

VisPro/Reports is the latest addition to the VisPro family of OS/2 programming products. VisPro users will recognize the easy-to-use CUA'91 user-interface and the drag and drop style HockWare pioneered years ago.

Designing Reports is Easy

- WYSIWYG, free-format report design tool.



Report Printing is Easy

- Banded report options allow you to choose from a rich set of headers, footers and groups.
- Label reports allow you to select from a large list of Avery labels.
- Support for multiple line text, derived fields, business graphics, bitmap images, rectangles, lines and ellipses.
- Many object properties to choose from, including any OS/2 font, 16 million colors, line styles, fill patterns and drop shadows.
- Wide selection of field options, International currency and formulas.
- Numerous sample reports included.
- Seamlessly print or preview reports from any OS/2 REXX environment including VisPro/REXX, VX REXX and plain OS/2 REXX.
- Use your favorite REXX-enabled databases and third party libraries for access to DB2, .DBF files, EHLLAPI mainframe connections and more.
- A runtime DLL of less than 100K can be distributed royalty free.

The Complete VisPro Family

VisPro/Reports	\$199
VisPro/REXX Gold	\$299
VisPro/REXX Bronze	\$99 \$ 59
VisPro/REXX Data Entry Object Pack ..	\$149 \$ 89
VisPro/C or VisPro/C++ SPECIAL!!!	\$399 \$199
VisPro Development Suite	\$599 \$499
(includes VisPro/REXX Gold, C and C++)	

Whether you've already discovered the VisPro family of visual programming products or you're just getting to know us, meet our latest member, VisPro/Reports. And be the first to get where you're going.

Special Offer
VisPro/C \$199
VisPro/C++ \$199
Save over 50%!
Offer Expires 8/15/95

HockWare
Putting You in Control

HockWare Incorporated
P.O. Box 336
Cary, NC 27512-0336
919-380-0616
919-380-0757 FAX
Go HockWare on CompuServe
hockware@vnet.net on Internet
<http://www.nando.net/ads/hockware>

HockWare, VisPro/C, VisPro/C++, VisPro/REXX and VisPro/Reports are trademarks of HockWare Incorporated. All other company, product and brand names are trademarks and/or registered trademarks of their respective holders and are mentioned for reference purposes only. © 1995 HockWare Incorporated. All rights reserved.

BEST SELLERS FROM THE Programmer's Paradise[®] CATALOG



Call for
a FREE
Catalog!

800-445-7899



c-tree Plus[®]

by FairCom Corporation

DOS • WINDOWS • NT • UNIX •
OS/2 • SUN • RS6000 • HP9000 •
MAC • QNX • BANYAN • SCO.

This well known, highly portable data management package has become established as the tool of choice for commercial development. Offering unprecedented data control, choose from direct low level access, ISAM level, or SQL access with the FairCom Server. Single User, Multi User, or optional Client/Server, ANSI Standard. Full Source. No Royalties.

Discount Price: **\$529**

Paradise No. FACT3100-UU



RimStar[™] Programmer's Editor

by RimStar

Technology, Inc.

Features: 32-bit GUI, Syntax Coloring, ANSI "C" macro language, "C" Source Browser, unlimited redo/undo, no limits on number of files, windows, or line length. Emulates BRIEF plus many other editors. Compile/jump to error, hex editing, smart indenting, bookmarks, completely configurable keyboard mappings, WF/2 enabled and much more. Windows & NT versions available.

Discount Price: **\$259**

Paradise No. RSRSP3100-UU



For more details
on the products
featured on these
pages call our
800 Number
and we will fax
you additional
information.



Gammathech REXX SuperSet/2

by SoftTouch Systems, Inc.

The SuperSet/2 gives REXX the power of C by providing interfaces to LAN Server, NetBIOS, TCP/IP and Communications Manager/2, and by providing access to low level system facilities, including processes, threads, semaphores, names, pipes, and VIO commands. The SuperSet/2 complements visual REXX programming packages. The 600+ page manual fully documents over 300 functions in 7 DLLs ready to supercharge your REXX!

Discount Price: **\$77**

Paradise No. GTGR33100-UU

HALO[®] Imaging Library[™] for OS/2

by Lifeboat Publishing

HALO Imaging Library for OS/2 enables you to incorporate powerful image processing into your applications with over 100 C/C++ imaging functions. This advanced toolkit provides a full range of functionality including the ability to display, enhance, print, manipulate, and save most popular digital image formats. Special effects such as sharpen, blur, warp, color conversion, scaling, and more are also included.

Discount Price: **\$716**

Paradise No. LPHILO3100-UU



Visual SlickEdit for OS/2—the intelligent programmer's editor by MicroEdge, Inc.

This high-powered programmer's editor is completely customizable with the speed to ignite your productivity. Save time using Brief, Emacs, or VI emulations. Take off with Visual SlickEdit's GUI interface, built-in dialog editor, and C-style macro language. Features include SmartPaste[™], compiler error processing, syntax color-coding, expansion and indenting, and an on-line manual. 30 day risk-free trial!

Discount Price: **\$279**

Paradise No. MEVSE3100-UU



DataTable for OS/2

by ProtoView

Development Corp.

DataTable is a sophisticated spreadsheet control. Through a robust message-based API and numerous notification codes, developers can control every aspect of their application. Features include: virtual memory management, column sorting, column locking, check boxes and combo boxes, colors and fonts may be set on a cell, row, column or table basis. New for OS/2, vertical window splitting.

Discount Price: **\$476**

Paradise No. PDDTO3100-UU



Prominare Validator

by Prominare

Prominare Validator works in conjunction with the OS/2 APIs providing additional

and detailed error information of your 32-bit OS/2 API usage. It provides details indicating which parameter you have passed to an OS/2 API that may be incorrect indicating exactly which parameter is in error. It also provides real time monitoring such that you can see these parameter errors as well as normal API errors visually. A truly time saving developer tool!

Discount Price: **\$92**

Paradise No. PRVAL3100-UU

OpenDoc &
OS/2 Warp
Development!

The Object Factory V3.0— for Warp & OpenDoc

by Synaptics, Inc.

The SOM development evolution continues! Introducing support for OpenDoc & SOM 2.1.

Features Multiple IR browsing, IDL importation, and method searching. OpenDoc class development is fully supported through the visual browser. Drag, drop, and compile. Visually browse and modify classes created with Direct-To-SOM compilers. Requires OS/2 Warp.

Discount Price: **\$240**

Paradise No. STOFW3100-UU



Programmer's Paradise[®]

1163 Shrewsbury Avenue
Shrewsbury, NJ 07702-4321

Call for shipping
charges/return policies.

Order Now! 800-445-7899

os/2 Developer

JULY/AUGUST 1995

THE MAGAZINE FOR ADVANCED SOFTWARE DEVELOPMENT



EDITOR'S COMMENTS
Objects, Notes, and PL/B 5



OS/2 TOOLBOX
IBM Developer Connection for OS/2 6



GUI CORNER
Printing Made Easy? 12



PROGRAMMING INSIDER
Some are from Mars, Some are from Venus 16



COMPILERS
PL/B—A Portable High-Level Programming Language 21
By William P. (Bill) Blocker, III



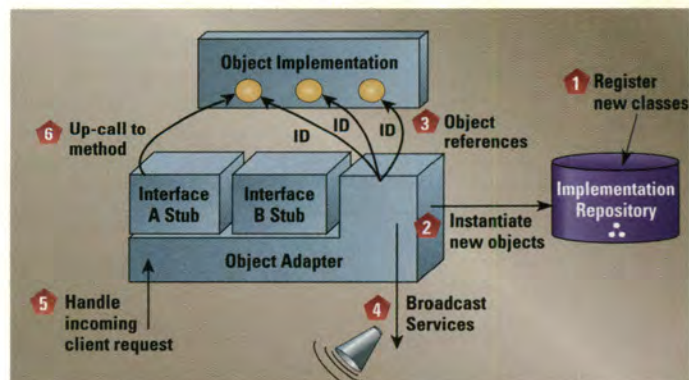
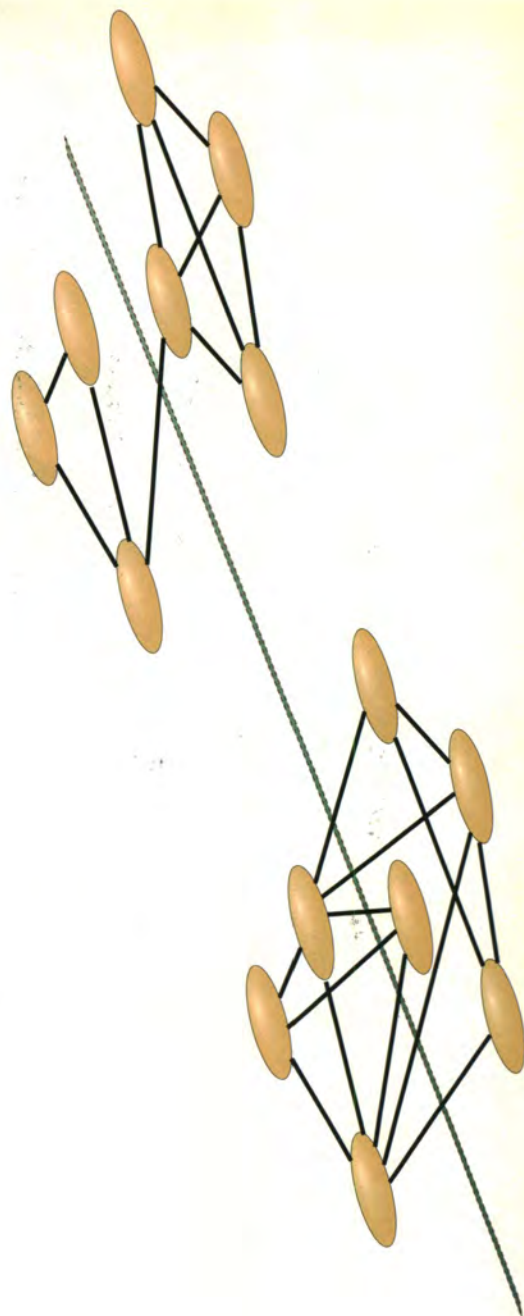
OBJECT-ORIENTED PROGRAMMING
The Server Side of CORBA 26
By Robert Orfali and Dan Harkey
Searching for a Cross-Platform Development Tool 31
By Maggie Sanders
Parts: The Newest Cottage Industry 34
By Rick Runyan
Distributed Objects: Application Design Tips 44
By F.R. Campagnoni, Ph.D.
Object-Oriented Tools Buyers Guide 57



SPOTLIGHT: LOTUS NOTES
Beginning Application Development with Lotus Notes 39
By Steve Kupec and Charles Carrington
A Lotus Notes REXX API 53
By Tony Walsh
Lotus Notes Buyers Guide 63



PRODUCT WATCH
New Products for OS/2 61



TRANSITIONING TO SMALLTALK TECHNOLOGY?
INTRODUCING SMALLTALK TO YOUR ORGANIZATION?
Travel with the team that knows the way...

THE OBJECT PEOPLE

Your Smalltalk Experts



SMALLTALK

Education & Training

VisualAge
IBM Smalltalk
Smalltalk/V
VisualWorks
ENVY/Developer
Analysis & Design
Project Management
In-House & Open Courses

Project Related Services

Object Immersion Program
Project Mentoring
Custom Software Development
Tools Construction

The Object People Inc. 509-885 Meadowlands Dr., Ottawa, Ontario, K2C 3N2
Telephone: (613) 225-8812 FAX: (613) 225-5943

Smalltalk/V is a registered trademark of Digitalk, Inc.
VisualWorks is a trademark of ParcPlace Systems Inc.

VisualAge and IBM Smalltalk are registered trademarks of IBM.
ENVY is a registered trademark of OTI Inc.

EDITOR Dick Conklin
CompuServe: 76711,1005

MANAGING EDITOR Marta Dils
mdils@mfi.com

EDITORIAL ASSISTANT Deborah Sommers

EDITORIAL ARTIST Peter Altenberg

VICE PRESIDENT/GROUP DIRECTOR
Regina Starr Ridley

PUBLISHER Peter Westerman
pwesterman@mfi.com

ADVERTISING SALES
Yvonne Labat
(415) 905-2353

Christian O'Brien
(212) 626-2322

Kristin Morgan
(212) 626-2498

MARKETING MANAGER Susan McDonald

MARKETING GRAPHIC DESIGNER Shari Flack

ADVERTISING PRODUCTION COORDINATOR
Glenn Sadin

CIRCULATION DIRECTOR John Rockwell

CIRCULATION MANAGER Stephanie Blake

SUBSCRIPTION INQUIRIES
U.S.: (800) WANT-OS2
Foreign: (708) 647-5960

un Miller Freeman
A United News & Media publication

CHAIRMAN OF THE BOARD Graham J.S. Wilson

PRESIDENT/CEO Marshall W. Freeman

EXECUTIVE VICE PRESIDENT/COO Thomas L. Kentp

SENIOR VICE PRESIDENT Warren "Andy" Ambrose

SENIOR VICE PRESIDENT David Nussbaum

SENIOR VICE PRESIDENT H. Verne Packer

SENIOR VICE PRESIDENT Donald A. Pazour

SENIOR VICE PRESIDENT Wini D. Ragus

VICE PRESIDENT/PRODUCTION Andrew A. Mickus

VICE PRESIDENT/CIRCULATION Jerry Okabe

OS/2 Developer has applied
for BPA membership.

PRINTED IN THE U.S.A.



BY DICK CONKLIN

Editor's Comments

Objects, Notes, and PL/B

In this issue, we've brought together several experts who are well versed in the everyday world of practical object-oriented programming. We are way beyond theory now; many of you are developing object-oriented applications and could use a few tips, tools, and techniques.

Frequent authors (you saw them here first, folks!) Dan Harkey and Bob Orfali are back to explain the CORBA Object Request Broker (ORB). ORB expert Dr. Frank Campagnoni adds some practical insight into the design of distributed object applications. Maggie Sanders describes her own real-life experience developing and maintaining her company's first cross-platform, object-oriented business application. And Rick Runyan speculates on the evolving use of preexisting parts to build applications from a technological and economic view. He should know—Rick manages IBM's new Object Connection program.

We've added a subtheme to this issue, a topic we've wanted to cover for a long time: Lotus Notes. Tony Walsh, author of *Lotus Notes Release 3 in the OS/2 Environment*, John Wiley & Sons, describes the REXX API for Lotus Notes and uses an interesting example: executing an OS/2 REXX script from a Lotus 1-2-3 spreadsheet macro. Lotus Notes application programmers Charles Carrington and Steve Kupec add a few tips of their own.

Can your OS/2 programming language create portable, object-oriented, cross-platform applications? Is it PL/B? What's PL/B, you ask? Bill Blocker tells us about one of those "best kept secret" tools.

IT ONLY LOOKS SMALLER

As we begin our eighth year of publishing for OS/2 developers, our magazine continues to evolve. High paper costs mean fewer pages, but changing to a smaller font size and putting more lines on a page has allowed us to maintain the same editorial content. Our authors are also uploading their long source code listings to our CompuServe OS2DF2 library. Our library and forum are great places to meet the authors, discuss the articles, ask questions, and find some excellent sample application code.

Our Managing Editor, Marta Dils, is leaving us for greener pastures back East. Marta has been the person who magically transforms a collection of rough articles into a beautiful finished product. We'll miss her. Good luck, Marta!

Dick Conklin



Dick Conklin

Contact	CompuServe	Internet
Dick Conklin (Editorial)	76711,1005, or OS2DF2 Forum	os2mag @vnet.ibm.com
Miller Freeman (Circulation)	76001,2317	mjohnson @mfi.com
Peter Westerman (Publisher)	76307,1054	pwesterman @mfi.com

OS/2 Developer: Vol. 7, No. 4, July/August 1995

Subscription information: U.S.: One year (six issues), \$39.95. Canada, Mexico, and international surface mail, add \$16 per year for postage. Canadian GST no. 124513185. International air mail, add \$30 per year for postage. Foreign orders must be accompanied by payments in U.S. funds. For new orders and customer service, call (800) WANT-OS2 (800-926-8672) or (708) 647-5960 (fax: 708-647-5972). IBM employees and branch office customers can subscribe to OS/2 Developer through IBM Mechanicsburg's Systems Library Services (SLS) using OS/2 Developer's order number: G362-0001. POSTMASTER: Please send address changes to OS/2 Developer, P.O. Box 1079, Skokie, IL 60076-8079. Allow eight weeks for subscriptions to begin. OS/2 Developer (ISSN 1073-0729) is published bimonthly by Miller Freeman Inc., 600 Harrison Street, San Francisco, CA 94107, (415) 905-2200. Application to mail at second-class postage rates is pending at San Francisco, CA, and additional mailing offices.

© Copyright 1995 by Miller Freeman Inc.



IBM Developer Connection for OS/2



Guy Scharf

What is your most important development tool? It's probably your application design tool, or your compiler, or another development tool. What's the second most important tool? Perhaps it should be the IBM Developer Connection (DevCon) CD-ROM subscription. I have found this the most valuable single source of tools and information available from IBM.

In 1992, IBM began issuing beta versions of a Professional Developers Kit CD-ROM. This CD-ROM quickly became the best source for the latest OS/2 programming toolkits. In early 1994, IBM introduced the Developer Connection subscription service. The service includes quarterly mailings of the Developer Connection CD-ROM and a 24-page large-size newsletter of OS/2 programming tips and techniques. The Developer Connection CD-ROM has grown from a single disk to three disks. The subscription service now also includes the Developer Connection for LAN Systems CD-ROM as well, so you get four disks each quarter. The latest DevCon for OS/2 is vol. 7; the LAN DevCon is vol. 3. A Device Driver Kit CD-ROM subscription is available as an upgrade to the Developer Connection subscription.

The CD-ROM is packed full of useful programs and information for the OS/2 developer. It includes toolkits, productivity tools, development tools and sample code, prerelease code, demos, product service packages, and extensive documentation. The documentation describes the license granted for each program. Programs are typically identified as demo, evaluation, internally developed, prerelease, reduced function, containing time-out code, or some combination of the above. Support for programs varies; some utilities have no support, other products are supported in electronic forums.

Most of the products on the CD-ROM are licensed for 120 days or until the next CD-ROM is released. However, licenses for these toolkits do not expire: MMPM/2, Pen for OS/2, Software Installer for OS/2, Developer's Toolkit for OS/2 2.1, Developer's Toolkit for OS/2 Warp, LAN Systems Toolkit, AIX DCE Toolkit, and the Encina Toolkit.

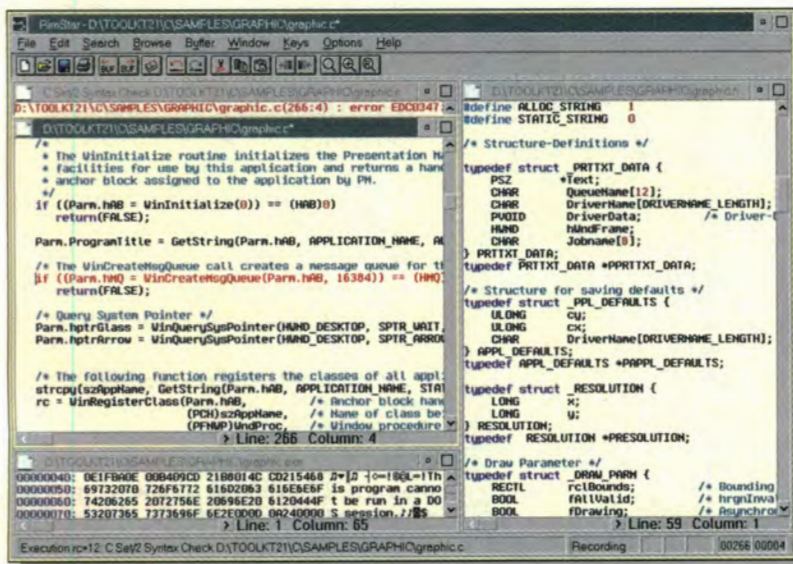
With each issue, some programs are added and others dropped. Typically, prerelease software appears for only an issue or two and is then dropped when the product is released. Release versions of other products may appear on one issue and then be dropped on the next, effectively granting you a free 120-day license to evaluate the products. An .INF file lists all of the products introduced with each issue, products that were dropped and the license terminated, and products that have been updated.

The installation process creates a Developer Connection folder and subfolders on your desktop. The two major components installed are the Browser, which allows you to search and read documentation, and the Catalog, which presents all of the programs and other files in an organized view from which you may review or install them.

Installation itself is simple. Insert any of the CDs, make the CD-ROM drive the current directory in a command window, and type `DEVCON` at the command prompt. The installation program offers you the opportunity to run from the CD-ROM or to install key Browser and Catalog files on the hard disk. DevCon requires from 95KB to 8MB of hard disk space, depending on the options selected. I always install all options and use the full 8MB on my hard disk. Otherwise, performance of the Catalog and Browser are severely degraded.

Once DevCon is installed, you can use the Browser to view documentation or the

POWER UP WITH THE RIMSTARTM PROGRAMMER'S EDITOR NEW VERSION 2.1



If you're looking for a fully configurable, professional OS/2 PM programmer's editor to replace your current text-mode editor - we have what you have been looking for!

No hassle changing editors

We know changing editors can be a major hassle. You don't want to relearn a new set of keystrokes no matter how good the product. Well, you don't have to - we have Brief, CUA, Epsilon, Multi-Edit, SlickEdit, PWB, and Borland IDE keyboard mapping waiting for you. If you're not using one of these, let us know and we will create the mappings you need.

Features designed to increase productivity

It has all the features you have come to expect, plus special features designed to anticipate your needs and increase your productivity. Features like a 'C' source browser that eliminates those time consuming searches for functions, global variables, typedefs, and macros by providing you with instant positioning to both definitions and references. Using our powerful 'C' macro language you can customize your programming environment to suit your individual needs.

Summer Special Price Only \$139.00*

Plus Shipping & Handling.

* Expires September 1, 1995

To order call 1-800-746-7007

60 day money-back guarantee.

Also available for Windows and Windows NT

Circle Reader Service Number 4

Partial Feature List

- ✓ Complete ANSI 'C' macro language
- ✓ SyntaColorTM—syntax highlighting
- ✓ Smart C/C++ indenting & brace matching
- ✓ Bookmarks
- ✓ Unlimited undo and redo
- ✓ Timed auto save
- ✓ Access PDK help for function under cursor
- ✓ Multi-threaded for no waiting
- ✓ Compile and jump to errors
- ✓ Import/export to system clipboard
- ✓ Keystroke record/playback
- ✓ Block indent/outdent
- ✓ Hex editing
- ✓ Customizable menus
- ✓ Column, line and block selection, search and replace
- ✓ Integrates with Workframe/2
- ✓ Source browser for 'C'
- ✓ Template editing
- ✓ Multi-buffer regular expression search and replace
- ✓ Support for version control
- ✓ Complete on-line help
- ✓ Save and restore state between sessions
- ✓ OS/2 2.x 32 bit PM Multi-document interface

"My copy of Brief has been permanently retired. Keep up the good work!" - A.L.

RimStar Technology, Inc.

91 Halls Mill Road
Newfields, NH 03856
Voice: (603) 778-2500
Fax: (603) 778-2408
BBS: (603) 778-4644

All products and company names are trademarks or registered trademarks of their respective holders. RimStar and SyntaColor are trademarks of RimStar Technology, Inc.

© 1994 RimStar Technology Inc.

Catalog to install programs. To install a program, start the Catalog, click on the + sign to open the list of programs, click with the right mouse button on the tool you want, and select Install. Installation of that product then proceeds normally. Figure 1 illustrates the Catalog and pop-up menu with options either to install a program or to view more information about it.

Most products can be installed directly from the CD-ROM. In a few cases, you have to create diskettes and then install from them. Fortunately, IBM is migrating to direct installation from CD-ROM for most products.

DEVELOPER TOOLS

If there's an OS/2 toolkit you need, it's probably here. Developer's Toolkits for OS/2 1.3, OS/2 2.1, OS/2 Warp, Pen for OS/2, OS/2 for SMP, MMPM/2, SearchManager/2, and more are here in their complete form. Kernel debuggers for different versions of OS/2 are provided, including ones for different Service Pak versions and for the Japanese version. An internationalization toolkit is also included.

The toolkits here are always the most recent available. I have found this the most reliable way to make sure I have the latest version of the toolkits. DevCon vol. 7, for example, has up-

dated toolkits for OS/2 2.1, SMP, and Warp.

The DevCon CD-ROM also contains tools for new technologies. The Developer's Toolkit for OS/2 for the PowerPC is included, as is the IBM PM debugger and Metaware's PowerPC Compiler. An OpenDoc Framework for C Set++ and an alpha OpenDoc Developer's Kit are also included, along with OpenDoc documentation and white papers.

Almost 50 development tools are included, both old and new. You should always look at the list of new and updated products with each issue. Sometimes there is real gold there! For example, DevCon 7 reintroduces a pre-release version of Validator for OS/2, which was last seen on DevCon 4. Recompile your program using Validator's header files and a compilation switch, and parameters to and return codes from all OS/2 APIs will be checked and reported in a separate window! This is an incredibly powerful tool, especially when working with a program that does not check API return codes thoroughly. Validator quickly identifies many API-related problems.

One Up Corp.'s SMART v.2 analyzes 16-bit OS/2 and Windows programs to assess the effort required to convert the application to 32-bit OS/2. SMART will place comments in the code indicating the conversions required and actually convert much of the code. Conversion is more complete for 16-bit OS/2 applications than for Windows applications but can be a very useful starting point in converting a Windows application to OS/2. If you need to convert an application to OS/2, this tool may be worth many times the cost of the DevCon subscription.

DevCon 7 includes a license for Software Installer/2. If you need to create an installation process for an application, Software Installer/2 is IBM's offering. For the IPF author, there are demo or reduced function versions of IBM's HyperWise 2 and Perez's IPF Editor. An IPF preprocessor allows you to create or maintain IPF source using symbolic values.

The CD-ROM contains demo versions of products from other vendors too. DevCon 7 has HockWare's VisPro/REXX Gold, Watcom's VX-REXX Client/Server Edition, and Quercus System's REXXLib among its development tools. Demo versions of both GUI and text editors are here, including CompuWare's Predictor/2, RimStar's Programmer's Editor,

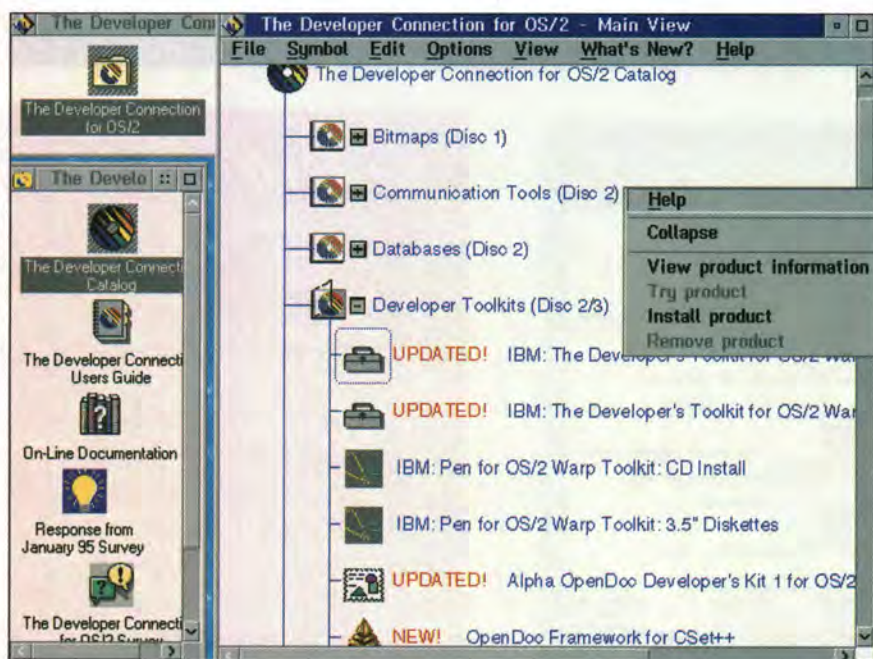


Figure 1. Developer Connection Catalog.

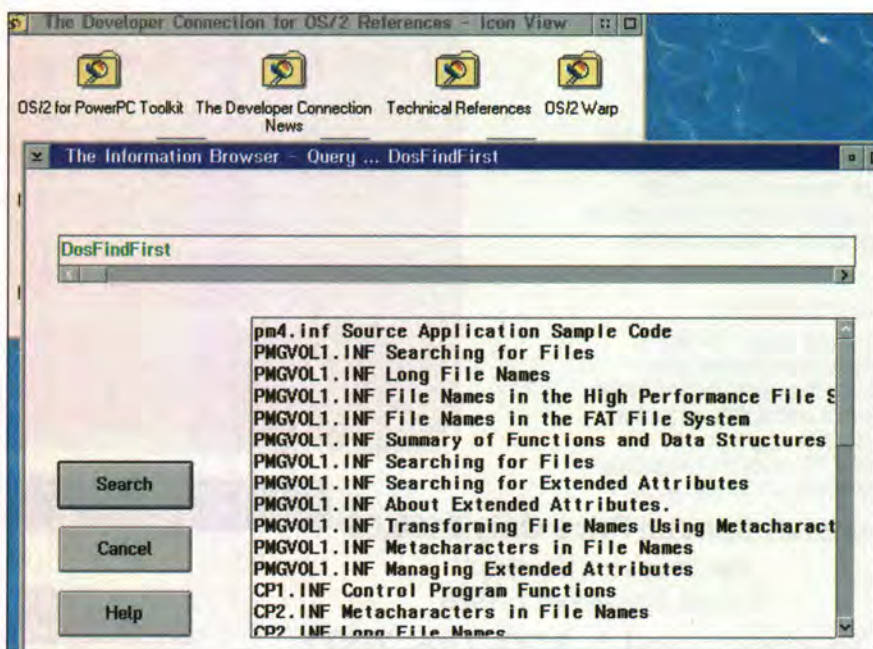


Figure 2. Developer Connection Browser.

MicroEdge's Visual SlickEdit, Boxer's Text Editor, and IBM's EPM. For the multimedia types, IBM's MMPM2 1.1.3 is included, along with a MMPM video poker game, Aria's Digital Music Player, and CD Explorer.

The Toronto Virtual File System (TVFS) is an installable file system that brings VM and UNIX concepts to OS/2. It supports links and search paths. This allows you to join different logical drives as a single drive or create an alias to another path within the file system.

There are a myriad of other utilities, including source code analyzers, debuggers, hex dump routines, and a toolbar control. IBM also uses DevCon as a way to deliver new versions of OS/2 and toolkits to developers quickly. When OS/2 Warp was released, IBM issued a DevCon 5 Special Edition to get Warp to developers. DevCon 7 includes a demonstration license version of OS/2 Warp and an updated BonusPak. Beta versions of OS/2, C Set++, and other products have been included in previous issues. A prerelease version of Object REXX will give you early experience with IBM's efforts to extend REXX to provide an object-oriented programming environment. ServicePaks and CSDs are included for C Set++, OS/2 2.11, OS/2 1.3 toolkit, TCP/IP, and WorkFrame/2.

In addition to the source code within the various toolkits, DevCon includes source code from all *Developer Connection News* articles and a selection of other source files as well. I have used the source files for a set of AIX-like utilities for OS/2 and for PMSPY32, a program that monitors and reports on messages in the PM message queues.

OTHER TOOLS AND UTILITIES

The DevCon CD-ROM contains a package of communications tools, including Communications Manager/2, Personal Communications/3270, LAPS, and even TCP/IP. Demo versions of Creative Systems Programming's Golden CommPass, OS/tools' COMi asynchronous serial device driver, a set of REXX communications APIs, and Softronics' Softterm Plus for OS/2 provide a range of communications programs and tools. If you need an Internet Gopher server for OS/2, one of those is here too.

For the database programmer, there's a time-limited evaluation copy of IBM's DB2/2 single-user version 1.2. Sybase includes its Open Client

Developer's Kit to allow you to write OS/2 applications that access Sybase.

In addition to development tools, the DevCon CD-ROM contains almost 40 "Productivity Tools." These programs include "IBM Employee Written Software" as well as commercial products. Some are useful to developers, while others may appeal to users. Of special interest to developers are these programs: PMTREE provides a graphical display of PM windows and allows you to interrogate or change window parameters. As a PM developer, I have relied on PMTREE for years. MAGNIFY is a desktop magnifier that allows you to examine your graphics at the pixel level. OS20MEMU shows you memory use of your applications. A timed, demo license copy of IBM's VisualAge for OS/2 gives the developer an opportunity to examine IBM's integrated application development environment designed for client/server applications using visual programming and allowing parts reuse.

Most of the end user products have demo licenses. ATS for OS/2 is a powerful job scheduling system. Applause is a TWIN-enabled image scanning and viewing utility for OS/2.

SpaceMap for OS/2 displays disk space use graphically. ViewAll makes it easy to view all of the books installed on the bookshelf. SAVEDSKF creates .DSK files from diskettes, and LOADDSKF restores .DSK files to diskettes. IBM's SearchManager/2 offers powerful text search and retrieval functions. And, just for fun, PM-GLOBE displays our planet Earth as a spinning globe, complete with night and day.

THE BROWSER

The Developer Connection for OS/2 CD-ROM includes more than 100 manuals. Some manuals are provided in OS/2's .INF format; others are supplied in Bookmanager's .BOO format. OS/2's VIEW program is used to read .INF format books. A Bookmanager read program is supplied for the .BOO files. A few files in PostScript—primarily files from Taligent—need to be printed with PostScript.

Open the Developer Connection folder and you'll see a folder labeled "The Developer Connection Browser." Open that folder and you'll see more folders. All of the documents are in there somewhere, often nested several



**Take control of your OS/2 scheduling needs with
Advanced Task Scheduler for OS/2**

September						
			1	2	3	
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

Are you looking for a way to manage your production job streams? Do you have programs that need to run after the completion of one or more other programs? Do you want to run programs on a regular basis? Do you want to process files that have just been received from another system or modified locally? Do you need to take action if a file is missing? Do you need to schedule around holidays and periods of heavy CPU load?

With ATS you can run your programs when YOU want them to run with out you having to be there.

Upgrades Available
from Version 2 and
other Task Schedulers
Call or E-Mail for Details

**Here are some of the features of
ATS for OS/2:**

- * Manage Production Job Streams
- * Programs can be scheduled to run anytime.
- * Programs can be dependent upon Files, Other Programs, Holiday Schedules, and External Signals
- * Integrated Operators Console
- * Logging - To File and Console

New Features:

- * Job Queues - for managing your resources while managing your tasks
- * Periodic Scheduling
- * COMPLETE API and Command Line Interface

Software and Consulting
2227 U.S. Highway #1 * Suite 146
North Brunswick, NJ 08902
Voice: (908) 821 - 0359 * Fax: (908) 821- 0350 * CompuServe 70312,627

Only
\$349

Circle Reader Service Number 5



levels deep. If you keep opening the folders, you'll eventually find the book you are looking for but will have missed the Browser in the process. Instead, select one or more folders, click with the right mouse button on a selected folder, and you'll see a new item on the pop-up menu: *Search*. Click on *Search* and you'll see the Browser dialog shown in Figure 2.

The search capabilities here are more powerful than in OS/2's native VIEW program. You can type complete phrases and the keywords *and*, *or*, *xor*, or *andnot*. And you can use parentheses to make more complex expressions. When you press the Search button, the Browser searches all books in the selected folders and prepares a list of matches. Searches of high-level folders can be quite time consuming as they may contain more than 100 books, even when the Browser is installed on the hard disk.

The Browser searches only documents in .INF or text format. To search Bookmanager documents, you must start the READIBM program and use its search functions.

DEVELOPER CONNECTION FOR LAN SYSTEMS

The Developer Connection for LAN Systems is documented as a separate subscription series but is included with the Developer Connection for OS/2. With only one CD-ROM, the contents are more limited. The LAN DevCon concentrates on documentation and product overviews. The only OS/2 toolkit is the LAN Systems Toolkit. AIX toolkits include DCE, Encina, and an evaluation license for SOMObjects. Evaluation versions of AnyNet, LAN Distance, LAN Server, NetFinity, NetView DM/2, and Person to Person/2 are included on LAN DevCon vol. 3, the current volume.

LAN product documentation is extensive and includes NetView DM/2, Encina, DSOM, DCE, MPTS, CID, LAN Distance, and LAN Server manuals. A variety of white papers, technical references, red books, and LAN systems marketing information rounds out the document collection. Some documentation is in OS/2's .IPF format, and other documentation is in Bookmanager .BOO format.

DEVELOPER CONNECTION NEWS

A Developer Connection subscription also includes the *Developer Connection*

News. This large-format newsletter is 24 pages long in vol. 7 and contains articles oriented toward the developer. Even the more general articles are technical in nature.

In the latest issue, articles discuss the evolution of operating systems from DOS to OS/2, the GRADD architecture for OS/2 for the PowerPC, Object REXX Scripting, an OpenDoc sample, and Visual Age++—the new name for what was called C Set++ in previous versions. Part 2 of an article on using SMART to migrate applications to OS/2 describes how to add your own migration rules. Other articles show a sample `SetupPrinter()` function for printer selection and discuss multithreading in graphics and Workplace Shell applications. Finally, there are two articles on writing device drivers. Source code for some of the articles is on the CD-ROM.

I have found the articles to be well written and to contain useful information about both standard technologies, such as multithreaded graphics, and newer technologies, such as OpenDoc and the PowerPC.

WARTS AND BLEMISHES

Technical Support for DevCon is provided on CompuServe on the OS2DF2 forum in a section restricted to DevCon subscribers. Support is also available on Internet and the OS/2 BBS. You may need one of these support mechanisms for later installations. DevCon installation runs well the first time you install a DevCon CD-ROM. But later installations can be more troublesome. None of the last several releases have had a trouble-free installation for me. With DevCon 7, for example, I had to manually remove all traces of the previous version before the Browser would install successfully. Other users have reported error messages during installation. With DevCon 6, installing the LAN DevCon would remove part of the OS/2 DevCon. All of these problems can be worked around, but the installation process needs improvement. IBM is trying, and each version gets a little better, but they're not all the way there yet.

The DevCon catalog provides a tree view of all of the programs on the CD-ROMs. The catalog includes the ability to print the current view of that catalog. Few developers are likely to find this feature useful unless reviewing the entire DevCon contents. That's fortunate, as the graphics printing

makes a glacier seem to move fast.

The Browser searches only .INF and text files. It would be nice if it would also search the Bookmanager .BOO files. I could also wish that the Browser was faster in searching documents. Even with the Browser installed on the hard disk, it accesses the CD-ROM when searching, and searches through many books seem interminable. The fastest possible CD-ROM drive is sure to help too. At least on my system, the Browser suffers from keyboard bounce. I have to watch what I type as keystrokes sometimes appear two or three times in the search string entry field.

Folders in the Browser are nested very deeply; the fully qualified paths to some objects get very long indeed. If you move the Developer Connection icon off the desktop and into a folder, the path lengths for some files may exceed the 256-byte system limit for fully qualified names.

SUMMARY

At \$199 per year (or \$119 for commercial and premier DAP members, students, and faculty), the IBM Developer Connection subscription is a "must have" for the professional OS/2 software developer.

Guy Scharf is president of Software Architects Inc., a software development firm specializing in developing OS/2 Presentation Manager applications for vendors and business. He can be reached at CompuServe 76702,557 or at Software Architects Inc., 2163 Jardin Drive, Mountain View, Calif.

PRODUCT INFORMATION

IBM The Developer Connection
for OS/2\$199/year

For commercial and premier DAP,
students, faculty\$119/year

Additional License ...\$75/year
For DAP\$45/year

Vendor: IBM Corp.
The Developer Connection
Internal ZIP 1599
P. O. Box 1328
Boca Raton, Fla. 33429-1328
Phone: 1-800-6DEVCON
Fax: 1-303-330-7655
Internet: devcon@vnet.ibm.com

Fast Visual Application Development for OS/2 and DB2



If you're looking for fast and easy application development for OS/2, then take a look at the award-winning Watcom VX•REXX visual development environment. VX•REXX lets you build applications to exploit the graphical user interface, multi-threading, and multi-processing power of OS/2. VX•REXX Client/Server Edition gives you the added power to access DB2 or other database systems, manipulate the

data, and chart the results at lightning speed.

"We like VX•REXX. Using it for development feels like driving a Porsche: it's fast, it's compact, everything's in the right place, and it makes us look good, too."

Peter Coffee, PC WEEK

Designed to Meet Your Needs.

Watcom VX•REXX combines a project management facility, visual designer and an interactive debugger to deliver a highly productive visual development environment. The Client/Server Edition includes additional powerful objects so you can rapidly create rich GUI database applications. You can create OS/2 client applications which connect to DB2/2 or DB2/6000. Use IBM's DRDA support on OS/2 to access DB2 for MVS, DB2/400 for AS/400, and DB2/VSE and VM (SQL/DS) for VM and VSE. Also supported are Watcom SQL and ODBC-enabled databases.

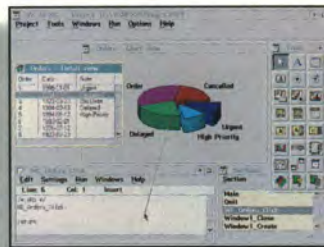
"Overall, this edition of VX•REXX for OS/2 is an outstanding visual client/server development platform." Nicholas Petreley, InfoWorld

- Over 2 dozen objects, including CUA'91 containers, notebooks, pop-up menus and more
- Easy to learn event-driven programming model with complete on-line documentation
- Support for professional multi-threaded, multi-windowed and drag-and-drop enabled applications
- Code reusability through section and file sharing
- Graphically create CUA'91 Presentation Manager objects, quickly customize their properties, and easily attach REXX procedures
- Package your application as an EXE or PM macro for royalty-free distribution

Point. Click. And Presto!

To create an application you draw user interface objects, customize their properties using standard OS/2 notebooks, and define their event code using powerful drag-and-drop programming. To add database access just draw a query object, visually design a SQL query, press OK and presto—your window is automatically populated with objects that are bound to your query to display, update and search your data.

"Drag-and-drop nirvana." Nicholas Petreley, InfoWorld



Give Your Data a Whole New Image.

Energize your applications by displaying your data in a 3D chart. The Client/Server Edition gives you more than a dozen chart types to choose from, along with over 150 display options. You also get complete support for run-time events so you can bring new drama to your data by making your chart interactive.

"VX•REXX is a must buy." Jacques Surveyer, ComputerWorld

Standard or Client/Server Edition— Which one is for you?

To start creating powerful OS/2 GUI applications right away, order your copy of Watcom VX•REXX Standard Edition for just....\$99*

Or, to start creating rich client/server database applications, order Watcom VX•REXX Client/Server Edition for just.....\$299*



1-800-265-4555

Watcom

A Powersoft Company

Watcom International 415 Phillip Street, Waterloo, Ontario, Canada N2L 3X2 Tel. (519) 886-3700 Fax (519) 747-4971 *Prices and specifications are subject to change without notice. Price does not include freight and taxes where applicable. Prices quoted in US dollars. ODBC drivers are available from INTERSOLV, Inc. Watcom, the Lightning Device, and VX•REXX are trademarks of Watcom International Corporation. Other trademarks are properties of their respective owners. ©Copyright 1994 Watcom International Corporation.

Circle Reader Service Number 6



Concluding the printing theme started with the last two issues, our intrepid cat-owned show illustrates how to get the screen to look just like the output on your printer.

By **MARK BENGE** and **MATT SMITH**

Printing Made Easy?



Mark Benge



Matt Smith

Okay, now where are we? We have shown you how to get the magical printer information (adendums included), shown you how to get the printer fonts and print out something using them, and have explained the wild and wacky terminology used to describe fonts. What else is there? How about getting the text in your window to match the spacing that comes out on your printer?

This process is trivial...if you know how. Well, if you don't, we are going to show you one of the ways of achieving this goal. Please note that we were just as much in the dark as you may be about this process. And we will warn you now, there are areas in which we are still in the dark. Nevertheless, let us begin our wondrous journey into the font and printer abyss.

The goal is getting a simple application that shows some text in the application window to provide a print preview. Also, the text in the window is supposed to match the spacing of the output from the printer. This is the real goal.

GETTING THE MEASUREMENT RIGHT

So where do we begin? First, how do we know if we have the right measurement on the display vs. the printer. Draw a scale? Yeah, right. Got any other ideas?

Why a scale? The idea is that we show a scale in inches on the display and on the

printer. When shown with the text, it will let us see if the text does indeed match. So how do we make sure the scale is the same between the two devices? We all know that the resolution of our printers is higher than that of the display.

Since our presentation units are in twips (that is, $1/1440$ of an inch), we can easily map our point sizes and other measurements. So what we do is make our screen presentation a permanent presentation space during our window creation using `GpiCreatePS` with `PU_TWIPS` as our presentation unit. When we go to the printer, we use `PU_TWIPS` as our presentation unit in the `GpiCreatePS` call. Now each of the presentation device drivers must figure out how to draw the scale appropriately. You may think this is all we have to do to make sure our fonts match, but this is not the case, as we will soon show you.

Our scale is really quite simple. We divide it up into increments of $1/16$ of an inch and make it look as close to a normal ruler or measuring tape scale as possible. Below this, just for effect and interest, we create a gray scale going from 100% black to 0% black in increments of 10%. Beside this, we show a color scale showing the major RGB colors.

WHAT ABOUT THE TEXT?

Okay, okay, onto the text. This is where things really start to get both interesting and messy. This problem almost made us want to go take power naps with the cats. But we digress.

First, we create a simple toolbar that shows the font face name and size. We use the code we developed from the previous column on selecting the font here. Once we have created our simple toolbar that contains the font and font size drop-down lists, we get our default font to use. This will be the first font within the font drop-down, so it will vary from printer to printer.

Taking the face name and size, we then create the fonts we will use to display the text. Here we have arbitrarily decided that

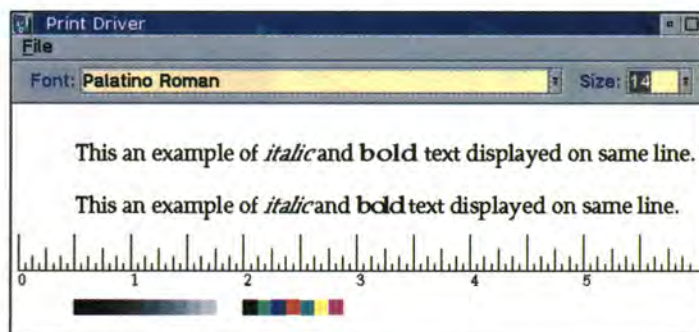


Figure 1. Printer Driver window.



we will be using three different types of font display: normal, bold, and italic. The text will be displayed so portions of the text will have the different attributes. The font selection routine has been designed with this in mind as have the storage structures. Nothing magical about this design exists; you can do it in a different fashion to suit your requirements. Also, after having gotten the first prototypes working, we decided to cache the font selections to achieve faster drawing of the text. You could have the drawing routine issue a `GpiCreateLogFont` each time a different font is needed, but the overhead for this is quite high, especially if you are reusing fonts. GPI is quite generous in that you can create 254 different fonts for use (this can be a mixture of font faces, styles, and sizes).

DON'T ALWAYS BELIEVE WHAT YOU READ

This section could also be titled "If you don't succeed, grab those books and manuals and try something else." When selecting the font, we first select it through the printer device context. This process lets us query the width table for the font. We do this because it is much faster to make one GPI call than many. The OS/2 documents suggest you use `GpiQueryCharString` to get the display points. This is a valid way of doing it, but you will also notice that the `GpiCharString` requires a vector of increments, not display points.

So, if you think about it, the suggested method within the OS/2 documents, while not wrong, tends to be a bit slow. Caching the width table lets you simply look up the character width for each character in your string and use this value as the vector width.

Once we have our width table, we discard the printer font selected and create the logical font for the display. Our font cache is going to contain all the information needed to create the proper vector array when drawing a text string. Also, we are going to create all the necessary logical fonts before we actually use them. This will mean that all the work will be done once and that the drawing will be as speedy as possible.

Here we use a trick suggested by Graham Winn in his book *OS/2 Presentation Manager GPI* (2d ed.) to create an array for the logical font IDs. When a new logical font is requested, the array is scanned for an opening, and the first available opening is used as the logical font ID. This value is then

cached in the font cache for use by the drawing routines. We do this for the normal, italic, and bold font.

Tales from the Trenches

Tale One

In the article "This Isn't the Gutenberg Press" (March/April 1995), we stated that the selected default printer should be queried using the `Prf` APIs. John Serences of DeScribe Inc. pointed out that a newer and better way exists. Using the `SpEnumQueue` API, you can determine the selected default printer by interrogating the `fsType` field within the `PRQINFO3` structure for the `PRQ3_TYPE_APPDEFAULT` flag. After digging through the reference manuals, we found information on this flag within the online PM Reference for OS/2 2.1 in the "Notes" section for the `SpEnumQueue` API. Also, we were notified that this flag was added to OS/2 2.1 and is supported in OS/2 Warp as well. We have updated our source code to reflect this newer, better way of determining the default printer when the sample app is running on 2.1 or higher. If we are running on 2.0, we still use the older method.

Tale Two

John also warned us of another problem that involves a conflict between the `Sp` and `Dev` APIs. The text of his note follows:

The `PRQINFO3` structure contains the field `pDriverData`. This points to a structure that contains the length of the private driver data and the private driver data. This driver data is the "default" job properties.

You might suppose that the length of this private driver is correct (and a constant for that printer). In other words, you might expect to get the same length back from a call to `DevPostDeviceModes` to retrieve the length of the job properties:

```
long size = DevPostDeviceModes (hAb, NULL,  
                                driver_name, driver_model,  
                                printer_name, DPDM_POSTJOBPROP)).
```

Such is not necessarily the case immediately after installation of a printer. The `DevPostDeviceModes` may return a number that is smaller or larger than that retrieved from `SpQueryQueue` until such time as someone opens the printer object, displays the default properties, and then O.K.s or cancels the dialogue. After that, the numbers will agree.

Now `DevPostDeviceModes` isn't a very friendly procedure. It doesn't appear that it uses the input parameter that indicates the size of the area you have reserved for it to output the private driver data. It returns the amount of data it determines you should have regardless of what you specify.

As a result, I recommend calling `DevPostDeviceModes` twice. Call it the first time to retrieve the size of the data area required. If this size is greater than the area currently reserved for the data, attach a new area, fill it with zeros, and move in the private driver data obtained from `SpQueryQueue` (or the last call to `DevPostDeviceMode`). Use the new area to call `DevPostDeviceMode` a second time to actually display the job properties dialogue.

Thank you, John, for supplying us with both pieces of helpful information, and to all, please keep those notes and suggestions coming.

IF YOU DON'T SUCCEED...

When researching the means of mapping the drawing of text of the display to the printer, we thought it would be neat if we provided the simple face names of the fonts you may have on your printer. By this we mean, if your printer has a set of fonts such as:

Courier
Courier Bold
Courier Italic
Courier Bold Italic

we would just show Courier within the toolbar font drop-down list. Then behind the covers, we would select the named bold and italic fonts.

Being the astute programmers we like to think we are (this is where things went higgledy-piggledy, folks), we read up on `GpiQueryFaceString` and thought our prayers were answered. Nope!

The call worked fine with Courier, but when we tried it on Helvetica, it just didn't want to cooperate. We kept getting the error `PMERR_FONT_NOT_LOADED` for the bold and italic fonts we asked for.

Hogwash! we said. The fonts are there, and when we enumerate the printer for them, they show up (don't you just love the cryptic error messages?). Not that we couldn't under-

stand the error message, it was just that no explanation was given as to what was causing the problem and how we could get around it. Also, we couldn't afford to wait until our old-age security checks came in before the technical support on CompuServe tried to answer this.

So what did we do? We went back to the books and manuals. Graham Winn's book talks about simply setting a type flag in the `FATTR` structure when selecting the font through `GpiCreateLogFont`. Not being proud, we did as he suggested and, lo and behold, it worked. We got our bold and italic font. (Just to be fair, we also used Graham's suggestion to use the `GpiQueryFaceString`, which didn't work the way he said.)

BUT DOES IT WORK?

Having gotten our font cache, now all we have to do is draw it. The drawing of the text is really quite straightforward. We select the type of font needed (normal, italic, or bold), build the vector increment array, and draw the text.

The easiest way to make sure we aren't pulling the wool over your eyes is to display the same text (same logical font ID) but without the vector

array. Here we simply adapt the drawing routine to bypass the building of the vector array and just draw the text. As you can see in Figure 1, the text on the top (this is the nonvector text) does not line up with the text below it.

And the printed result? Try it and see. Also, try measuring the scale—it matches a real ruler! Now, try doing that with the scale on your display.

If you select a font on the printer for which you do not have a corresponding screen font, you will see unpredictable results. Usually, you will get back the system font for the selected logical font.

VIEWING THE SOUL OF YOUR OUTPUT

Implementing the print preview is really easy once we have all the other drawing support in place. The basic way of doing it is to create a bitmap image of the page to be printed and then scale it down to the size of a postage stamp (well, not that small, but you get the idea).

What we do here is create a memory bitmap that corresponds to the actual size of the page that would be printed. The only trick we want to warn you about is that all our drawing is in twips and the bitmap is sized in pels.

You may think that this is going to throw things out of whack, but it won't. You only need the dimension of the bitmap image in pels for the `BITMAP-INF02` structure (this was one of the times we really wanted to take a power nap with the cats to forget our woes until we figured it out). You can still draw in the units you have selected for the PS whether they be twips or some other system of measure that `GpiCreatePS` offers. OS/2 does all the work of converting things from our selected measurement system to the actual device measurement.

All we needed was the page size in pels. `GpiConvert` does the work of converting the size of the page in twips to pels. We use these values in the `BITMAP-INF02` structure for the `cx` and `cy` of the bitmap. The drawing routines we use to display the text and scale within the application window and on the printer are called this time using the bitmap presentation space.

The dialogue displayed in Figure 2 takes the full-size (1:1) bitmap image and through `WinDrawBitmap` draws it to the appropriate size in the `WM_PAINT` message processing. This is done once we let the dialogue paint itself. `WinDrawBitmap` scales the preview bitmap to the appropriate size for us. It really is that simple.

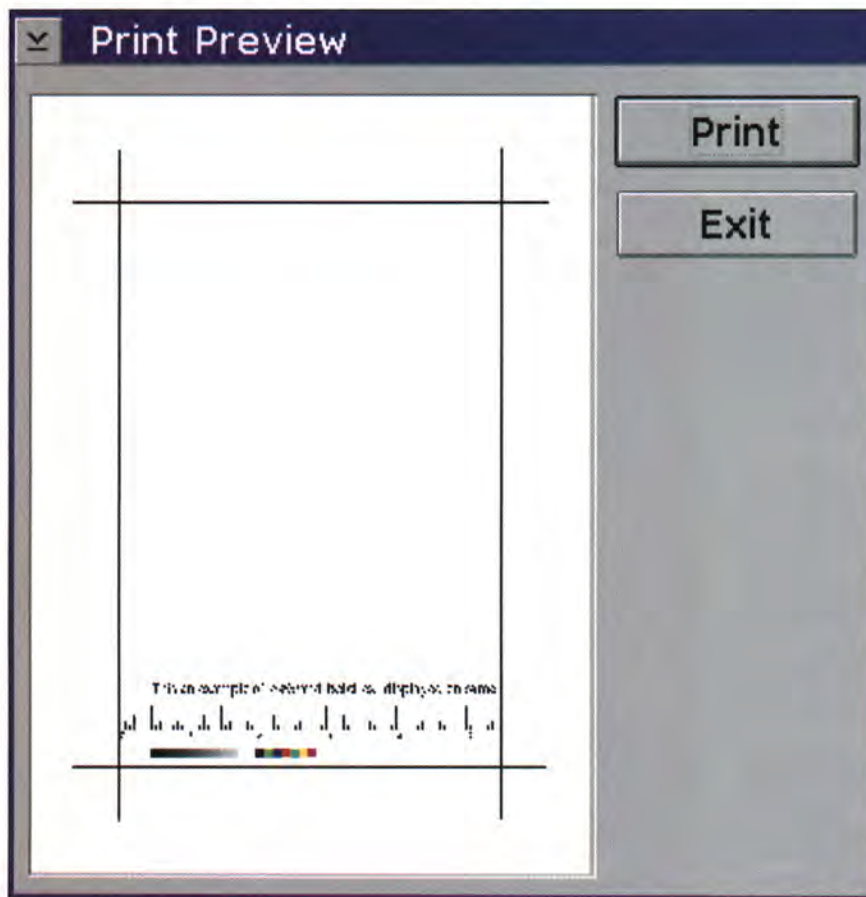


Figure 2. Print Preview dialogue.

SETTING UP THAT PAGE

With all this done, wouldn't it be nice to have a page setup dialogue where you could select the printing form to use along with the page margins, header and footer? No problem. Well, maybe just a little, or a lot. Figure 3 shows the Page Setup dialogue in its final form. Early editions had a drop-down list that allowed the selection of the form to use. Here you would use the value selected for the form within the spooler parameters. Nice idea, but there are problems.

The biggest problem was that this isn't supported consistently by the printer drivers (interestingly, during our research, we were told that one printer driver that shall remain nameless wasn't doing it right and, aha, a problem was registered against it; so be warned, don't try it on the PostScript driver). The only method we could find that would work was to let the user cause the job properties to be displayed and then have the form selected from the dialogue.

As for the margins, it was a piece of cake. Here we let the margins be entered, and these values were then used to provide the necessary clipping rectangle. It was easy. Just multiply the margin value by 1,440, and you have your margin ready to use.

REFERENCES

Credits: Screen captures were done through OpenShutter from One Up Corp. and touched up through PaintBrush in Win-OS/2.

Special thanks to Matt Rutkowski of the IBM OS/2 Print/Spool Development Group for providing insight into the printing. We would also like to thank Graham Winn for providing further secrets from the dark.

You can download PRNT3.EXE from these electronic sources:

CompuServe's OS2DF2 forum (OS/2 Developer Mag Section).

File area 11 on the IBM PCC BBS, (919) 517-0001, OS/2 Tools section on the OS/2 BBS for IBM TALKLink customers.

FTP to the site address prominare.com and log in as anonymous. The source is located in the /pub/prominare/guicornercodecache subdirectory.

Also check out the new GUI Corner home page at www.prominare.com/prominare/guicorner.html.

Again, we show a preview image, but this time within a group box. We use the group box to frame the image and determine display points for the image.

WHAT ABOUT...

You may have noticed that we haven't provided users with the ability to select the orientation of the printout in any of the dialogues. Why?

Unfortunately, there is no programmatic method to select portrait or landscape. The only thing that can be done (sometimes successfully and sometimes not, depending on the printer driver) is a rotation transform on the printed output.

Why not use this all the time? Well, you have problems with fonts, specifically device fonts that may not be able to be rotated. Also, bitmaps sometimes don't handle the rotation transform. So we thought better of showing it because if it doesn't work consistently, you really can't use it day to day on different printers.

FURTHER SOURCES

If you want a good book that shows some really neat font handling, check out Graham Winn's *OS/2 Presentation Manager GPI* (either first or second edi-

tion). Both editions really have some interesting gems within them. The new *OS/2 Warp Technical Reference Library Graphics Programming Interface Programming Guide* has a good discussion on both fonts and printing. *The Real-World Programming for OS/2 2.11* (2d ed.) by Blain, Delimon, and English has a good discussion on fonts and a section on printing.

Mark Benge, IBM Software Solutions, Research Triangle Park, N.C., is a staff programmer who joined IBM in 1989 and has worked on various CUA '91 controls for OS/2 2.x. He works in IBM open class library development, where he was involved in the implementation of C++ classes for drag-and-drop in C Set++ 2.1. Mark has a B.S. in computer science from Western Carolina University. He can be reached on CompuServe at 73532,2063 or via Internet at banzai@vnet.ibm.com.

Matt Smith, Prominare Inc., Toronto, Ont., is lead architect for the Prominare Development System, an OS/2 2.x advanced GUI development environment. Matt has been actively involved with OS/2 since 1988 and cofounded Prominare in 1990. He has a degree in architecture from the University of Waterloo. He can be reached on CompuServe at 70363,1175 or via Internet at msmith@prominare.com.

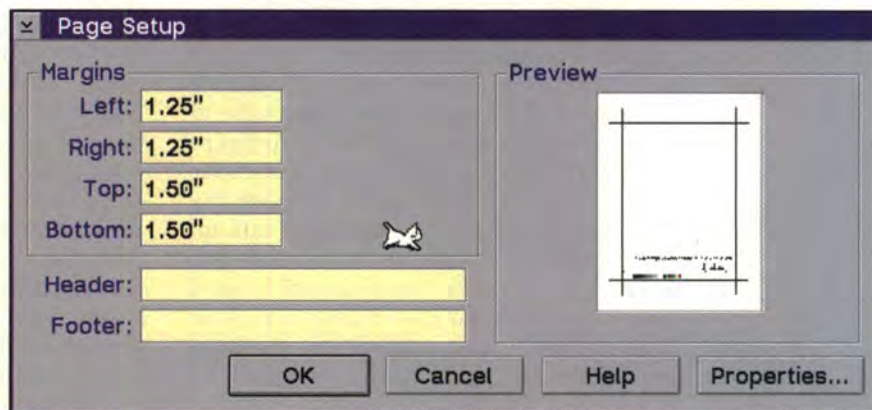


Figure 3. Page Setup dialogue.

GUI Corner on the Internet

You can now easily check out the source code offerings from GUI Corner through its new Web home pages. One of the home pages, <http://www.prominare.com/prominare/guicorner.html>, provides an easy way of selecting current and past source code examples.

A second home page, <http://www.prominare.com/prominare/guicorner.catsmeow.html> contains, well, ah, umm, errata, shall we say, for both our source code and column. Yeah, we make mistakes and will admit it, too.

Also, some of you have better ideas than we do or take our ideas and extend them. This area is to share the fixes and ideas. We will also include suggestions and other comments from you.



Getting some people to talk to each other can sometimes be an impossible task. Getting your programs to talk to each other doesn't have to be. In this issue's Programming Insider, David Reich explores the different interprocess communications constructs available to you and shows you how to pick the one that is right for you.

By DAVID E. REICH

Some are from Mars, Some are from Venus



David Reich

If you've read or at least heard of the book upon which the title of this column is loosely based, you know it is about communications. In the context of OS/2, communications is an interesting topic, to say the least. I'm not talking about communications programs here, but applications communicating with each other.

You know that OS/2 runs DOS and Windows programs as well as OS/2 full-screen and Presentation Manager-based graphical programs. Each has communications facilities available to them and, as with most things in OS/2, there are many ways to perform similar tasks. Your job is to figure out which ones are appropriate given your technical needs, taking into account the type(s) of program(s) you wish to talk to. Once you have decided on the methods you want to use, you need to look at how (or if you even need) to coordinate access to shared structures or coordinate message passing.

WHAT'S AVAILABLE

Inasmuch as OS/2 protects applications from accidentally stepping on each other, it can be a daunting task to use the architected methods when you really want them to share data. You have Dynamic Data Exchange (DDE), both named and anonymous shared memory, the clipboard, pipes, and queues, along with coordination tools such as semaphores and must-complete sections. Then, of course, there's always the homegrown mechanisms you may come up with. With so many to choose from, how do you know what to use, or even what you can use? That's what we're going to do here.

The first thing you need to do is look at your program and its attributes (is it OS/2 full screen? PM graphical?) and the programs with which you want to share data.

For example, do you want to share data among only parts of the same application? Do you want to be able to share data with Windows programs? Only OS/2 graphical programs? Any program that may be running in the system? These decisions form the foundation and must be made before you go any further.

Once you look at these attributes, you must next look at the structure of your program. For example, if all you need is to share data among threads of your application, you do not need to take any special action because any thread within a process has access to the process's data. That is the simplest case. How about the situation where you have an application that utilizes multiple processes? The example I like to use is a program for a medical facility, where the task is to capture and display heart rate information for analysis. First, why would you want to implement this as a multiprocess program rather than just have threads to gather data and others to log the data and display it?

The reason is protection. The critical part of the overall application operation is data gathering and logging. The display is secondary. The most important part is not losing any data. If this were implemented as a single process, a display thread exception could take down the entire process and cause the loss of data.

By separating the application into a gather/log process and a display process, you can reduce the exposure. (Of course, if the logging process goes down and takes the gathering function with it, it is of no matter. If that were a separate process and it went down, data would be gathered and go nowhere anyway. Hence, we can put them in the same process.) You now see that data sharing includes single applications as well as different ones. In this case,

An Internet-based Directory to Thousands of Products & Services

OS/2® & AIX developers, systems administrators,
and users: the tools you need are just a few
keystrokes away.

The best way to find out what's available
for IBM system software platforms.

Stay on top of it by browsing
OS/2 Developer's software and
services directory of products
for OS/2®, AIX, and other
IBM supported

operating
systems.

Comprehensive . . .

Thousands of listings,
multiple hardware and software
platforms, more than 30 industries, and
over 100 different product categories.

Current and Regularly Updated . . . Miller
Freeman, Inc., high tech publishing experts and the
publishers of *OS/2 Developer*, update the information
continually, so you have access to the most cutting-edge
tools as you need them.

Quick & Easy . . . The super fast search engine lets
you look for just the right product or service.

Find It
on the
Web!



Sponsored by IBM®,
OS/2 Developer, *OS/2 Magazine*,
and *Software Development*.

OS/2® is a registered trademark of
International Business Machines
Corporation and is used by Miller
Freeman, Inc. under license.

getting data to the display routines is no different than sharing data among differing applications.

Let's now go through some of the questions that must be asked when designing the communications mechanisms. You already know what type of program you are writing, and you already know what other types of programs (or even your own) you want to communicate with. Each communications technique has a set of properties that suit it for specific tasks.

CLIPBOARD

The clipboard is essentially a piece of shared memory inside the operating system. There is only one of these, and it holds predefined data formats. It can hold text data, metafile data, or an application-defined data format.

Following are some of the properties of the clipboard: it holds only one block of data; when another block is written into it, the previous block is overwritten; reads and writes are usually user-initiated; and it (or, more pre-

cisely, the data inside it) can be shared among Windows, OS/2 graphical, and windowed (DOS or OS/2) applications. Use of the clipboard also requires no special coordination or semaphore protection because the clipboard APIs handle that for you.

DYNAMIC DATA EXCHANGE

DDE can also be used between Windows and OS/2 graphical programs. Since it is a Presentation Manager facility, windowed DOS or OS/2, as well as full-screen programs, cannot use DDE. DDE is usually user-initiated to some degree (such as when data is marked for communications) but does not have to be.

Coordination with DDE is accomplished with the use of the DDE messages; and data can be sent through the messages, or the messages can point to shared memory areas. The sending (server) and receiving (client) applications, or processes, must know about the topic to be "discussed" in the DDE conversation as well as the data formats, as they are not predefined. While DDE is extremely useful for ongoing data exchanges (the "hot-link"), it can be used for one-time exchanges as well.

SHARED MEMORY

Shared memory is exactly that. In general, a single block of memory cannot be accessed by more than one process, but OS/2 has a construct called shared memory that allows multiple processes to access the memory object. This is a low-level structure in that it is just an accessible block of memory. While DDE and the clipboard have coordination mechanisms in their messages and APIs, respectively, access to shared memory must be carefully coordinated with semaphores or other constructs, such as must-complete sections.

Shared memory can hold any type of data, and like DDE, the writer and reader must understand the same data formats for it to be useful. This construct cannot be used to share data with DOS or Windows programs. Shared memory is not usually user-initiated as such (have you ever seen a menu choice in an application that says "Put data into Shared Memory"?), but user action can indirectly tell an application to create or use some shared memory.

There are two flavors of shared memory, the named shared memory and the anonymous shared memory. The difference between the two is how and which programs can obtain access

Quadron development tools for IBM ARTIC cards:

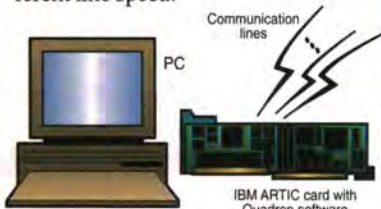


How to communicate better.

Increase system efficiency.

Our development software allows you to shift communication tasks from a PC to an IBM ARTIC co-processor card specially designed to handle them.

You'll gain CPU independence and multiple high-speed ports. Plus, each port on the IBM card can potentially support a different protocol and a different line speed.



Choose just what you need.

We have off-the-shelf solutions for your everyday data transmission needs, and tools for custom situations. You can select from HDLC/SDLC, Bisync, Async, X.25, LAP-B and custom protocols.

Your host PC can be an ISA, EISA, or MCA machine running various operating systems. And possibly most important, you can port co-processor code that you write from one operating system to another with little or no modification.

Serve vertical markets.

Quadron and IBM have teamed up to help people worldwide manage demanding applications such as:

- Travel reservation
- Stock market data delivery
- Telecommunications switching
- Retail point-of-sale purchases
- Process control
- Automatic teller services
- Shop floor control.

Work simpler, work smarter.

Quadron software is easy to install, easy to use, and downright productive. The high-level, C-language API makes co-processor programming simpler. Our user manuals have numerous code examples to speed you on your way. And our support, should you need it, will quickly serve up accurate answers to questions, whether they come in by phone, fax, or through our BBS system.

Act right away.

For today's dependable solution to tomorrow's communications needs, contact us now.



Quadron

209 East Victoria Street
Santa Barbara, CA 93101 USA
fax 805-966-7630
telephone 805-966-6424



© 1995 Quadron Service Corporation. All trademarks are the property of their owners.

to the memory. If named shared memory is used, all participating applications must know the name of the shared memory. Other constructs can be used to communicate the name to other programs, but this involves even more work on your part. Anonymous shared memory does not require the applications other than the creator of the memory object to know it before requesting access to it.

QUEUES AND PIPES

Queues and pipes are also low-level constructs (they do not have built-in coordination or data formats) just like shared memory and share many of the same properties of shared memory. Shared memory potentially can have many readers and writers (assuming you can juggle the coordination properly), as can queues and pipes, but the latter generally are used for one or more writers and one reader.

Named pipes can be used to share data with DOS programs, but not without a significant amount of work. Queue data cannot be shared with DOS or Windows programs running under OS/2.

THE ANALYSIS

You can see from my descriptions of the interprocess communications constructs where this analysis is leading. The best way to illustrate this is with the example outlined at the beginning of this column: the medical facility program. While it is impractical to go through all scenarios here, this one will show you how to choose the right one (or at least the one you like the best) for a particular situation.

Recall that this will be a two-process application. (You can look at these as two separate programs, which they are!) One process communicates with data analysis equipment and gathers time-critical data. This process has threads for data collection and threads that log the data to a file or some other database. The second process displays the data in some format on the screen. Let's look at the properties and parameters we have to work with.

First, we have data that is probably not in a standard format. We could use text data, but that means code to interpret the text into meaningful information, doing a lot of ASCII conversion to numbers, and so on. So we can safely say that even though we haven't written the code yet, the data is not text or metafile. There is also no concern currently about communicating

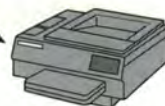
with Windows programs, but it would be nice to not preclude this function if at all possible.

Next, do we want the data as an ongoing exchange or a one time transfer? By the very nature of the task, it must be ongoing. It might be useful to allow the user of the application the ability to start and stop the display from changing, so a menu choice (or a button) should be able to stop and start the data transfers. Now that you have the parameters defined, you can

map them to the capabilities of the various IPC constructs.

I think you'll agree that the clipboard is not a good choice. You'll probably want an ongoing data exchange, and the clipboard can't do that without extensive programming and a significant amount of overhead. You could program the application to automatically write a set of data into the clipboard every so often and then have the reader get it, but then you have coordination issues to deal with, as

SINGLE KEY-STROKE SCREEN PRINTING



CLIPBOARD
VIEWER



SCREEN
SAVER

SCREEN CAPTURE
.WPG .MET .BMP .PCX .TIF .GIF .PCT



- Quality Color or Black & White copies of color Desktop Images.
- Copy any portion of any image on the Desktop.
- Screen Saver prevents monitor burn-in.
- Date & Time Display for Time-Stamping images.
- LAN installable. Complete support for LAN attached printers.
- Economical! 4 Utilities for 1 Price! Entity licensing available.
- The most stable and reliable product available.
- Includes both OS/2 2.1/2.0 (32-bit) and OS/2 1.3 (16-bit) versions.

THE LEADER: PrntScrTM Screen Image Utility for OS/2



TM

MITNOR Software

Phone: (918) 357-1628

Fax: (918) 357-2869

SCREEN IMAGES IN OS/2 DEVELOPER ARTICLES ARE CAPTURED USING PRNTSCRN, COURTESY OF MITNOR SOFTWARE

Circle Reader Service Number 7

you'd be using the clipboard in a way that is not intended. The clipboard is mainly for user-driven exchanges.

So the clipboard is out. How about DDE? It has free-form data, ongoing communication ability, a way for users to initiate and terminate it (via buttons or menu choices that you provide), and a defined communication protocol with synchronization. Both processes need to know about the data format, but that would be the case with any IPC mechanism you use for

this application. DDE is also possible between OS/2 and Windows programs, so flexibility is there if you want to bring a Windows program running under WINOS2 into the picture later. So why would you want to use DDE as opposed to a pipe or queue?

DDE, pipes, and queues provide many of these same features, but I would choose DDE in this case for two reasons. First, DDE provides the communications protocol as well as the

synchronization in that protocol. Call me lazy, but if the system provides it for me and I don't need anything else that it does not supply, I'd rather use it than start having to deal with semaphores, full and empty pipes and queues, and so on. The second reason is that queue data cannot be shared with DOS or Windows programs, and although data can be shared with named pipes, it seems that it is more work than is necessary, especially given that DDE handles this already.

There are plenty of reasons to use pipes and queues over DDE, but not in this case. One reason might be if you have variable-sized data, or if you are passing different types of data through the same structure to multiple processes. You can see how you need to analyze the functions that OS/2 offers to make the appropriate decisions for your applications, not only to avoid confusing your users but also to make your application development faster and easier. Complicated is not always better.

You can map this analysis to your applications that communicate with other things you have written as well as those that communicate various types of data, such as text and metafiles. This is just one more consideration in your analysis.

Choose the communications tool that best suits your needs without forcing too much work on you. Take the attributes of the various IPC constructs and map them to the fundamental needs of your applications and you'll have data exchange that is efficient and robust. Just because some are from Mars and some are from Venus does not mean they can't talk to each other effectively.

David Reich has been with the IBM OS/2 development team since 1987. He has worked on many parts of the system, supported customers and application developers, and traveled the world giving seminars and teaching OS/2. His latest book is *Designing High Performance OS/2 Warp Applications*, published by John Wiley & Sons. He can be reached on CompuServe at 76711,632 or via Internet at speedracer@bocaratton.ibm.com.

**Vision is far more than
just seeing.**

**With knowledge it
becomes understanding.**

**With understanding it
becomes power.**



**Come -
Soar With the Eagles
ColoradOS/2 1995**

**October 15-
October 20, 1995**

**Keystone Conference
Center
Keystone, Colorado**

ColoradOS/2 is a production of
Kovsky Conference Productions, Inc.
<http://www.colos2.com>

**OS/2 Programming
Summit!**

ColoradOS/2

This is Western-style fishing, as illustrated by the great American Bald Eagle. Relying on his intelligence and his incredible vision. In the wild, vision can be the difference between having lunch, and being lunch.

In our high-tech world, superior vision can be the difference between "Let's do lunch!" and "I just had my lunch eaten."

Of course, vision is far more than just seeing. You must apply wisdom, judgment, knowledge and experience to make sense of what you are looking at.

At ColoradOS/2, we supply knowledge to those software developers whose superior vision has led them to OS/2. All you add is wisdom, judgment and experience.

Oh, and we supply lunch!

**Plan now to
attend the Fourth
International
ColoradOS/2
!!!**

**To register or for information call:
(800) 481-3389 US & Canada
(719) 481-3389 International
(719) 481-8069 FAX**

OS/2 is a registered trademark of International Business Machines Corporation.

Circle Reader Service Number 8



This article discusses a high-level programming language that is easy to learn and may be more prevalent than you think. By WILLIAM P. (BILL) BLOCKER, III

PL/B—A Portable High-Level Programming Language

A business or organization has many needs to consider when choosing a programming language for its computer systems. The days of a single-platform, single-data-format shop are gone. Businesses and their programmers are tied to one key word: productivity. The time they spend writing and maintaining the programs that are the very lifeblood of their businesses simply does not allow for complex or wordy programming languages.

High-level languages were developed to improve programmer productivity and overcome potential errors. While CASE tools are a blessing to productivity, they can be abused and result in programs that create unnecessary overhead and perform poorly on any operating system, even OS/2. CASE-generated code often requires fine-tuning.

PL/B—A LANGUAGE WHOSE TIME HAS COME

The computer world is only just now really focusing on the power of OS/2 (not just as an alternative to DOS and Windows, but as the next generation in operating system power). Programmers now more than ever need the power of a language such as PL/B (Programming Language for Business). PL/B compiler developers (listed at the end of the article) have ported, or are porting, their products to OS/2—not just as shell products, but as new native language compilers for OS/2. What makes PL/B even more valuable is that it is a mature and fully developed language recognized by the U.S. and other governments, as well as Fortune 500 users who have written hundreds of thousands of applications—programs that need only to be enhanced to take full advantage of OS/2.

An impressive list of organizations use PL/B, but one reason for its obscurity is that many of its users restrict PL/B compiler vendors from referencing their companies in ads

and marketing brochures. (Note: while I'm bound by these same restrictions, let me say that most readers have been served by PL/B applications when buying a computer or new car, getting a car repaired, renting a car, or undergoing a tax audit—to name a few applications.)

PL/B AND OS/2—MADE FOR EACH OTHER

The histories of PL/B and OS/2 are quite similar in that both have roots in multitasking and the potential of exploiting the power of multitasking, multiuser, open client/server systems. OS/2 provides the platform and PL/B provides a way to tap its power. Even non-object-oriented programs written for and running on older operating systems can be ported to OS/2, compiled, and used as they have been. Later, these programs can be OS/2-enhanced to their full potential. When minor changes are needed, such as for new file-naming conventions, multitasking, or GUI enhancements, they can be addressed by compiler directives. Application programmers maintain a single copy of source code for a variety of computer systems. PL/B programmers don't convert code, they enhance it for new platforms. Such enhancements are often transparent to the source code, which is simply ported to OS/2 and compiled. Other enhancements can be added as time and money permit.

BUT WHY PL/B?

Most PL/B compiler developers started as porting specialists for businesses faced with changing computer platforms but wishing to avoid new code generation and retraining of their personnel. Porting and writing applications in PL/B for OS/2 immediately offers:

- True 32-bit, GUI, object-oriented programs
- Virtually unlimited control of each piece of data right down to the byte level (or

even binary digit control—a slight overkill); one of PL/B's major strengths is its string manipulation

- Simultaneous access to data from AS/400, VAX, RISC/6000, Novell, and other servers via SQL or ODBC
- Access to ASCII file data faster than DB/2 or other traditional database access methods, thanks to a unique accessing mechanism known as AIMDEX
- Programs that don't destroy data if power is lost; PL/B performs "logical" instead of "physical" file opens that other languages use
- Programs that can efficiently support single or multiple users (even on the same OS/2 machine) through serial ports, with or without special device drivers.

PL/B—ITS ORIGINS

PL/B originated in the 1970s when Datapoint Corp. (the inventors of ARCNET) introduced a high-level business application computer language known as DATABUS. For many years, this language was restricted for use on Datapoint computers only. However, the users of this language, even though some were often forced into it by their employers, developed a loyalty seldom found in any other language—a loyalty that surpassed the computer for which it was originally developed. Hence, a number of software firms began developing what was to become PL/B for personal computers and many other platforms.

Based more on the strength of the language and loyalty of its users than on public exposure, PL/B (then called DATABUS) has been standardized by an Accredited Technical Committee of the American National Standards Institute). In 1994, ANSI Technical Committee X3J15 officially renamed the DATABUS language to PL/B. Their work continues to expand and enhance PL/B standards to address the needs of the 1990s and beyond.

WHY ABANDON YOUR CURRENT LANGUAGE?

At this stage of an article, many authors would launch into a comparison of their favorite programming language(s) vs. others such as COBOL, Visual Basic, C, C++, or whatever. This one-upmanship approach is unnecessary and irrelevant with regard to PL/B.

To be honest, I respect (but by my choice would never use) COBOL; it is a 50-year-old language, and there are literally billions of lines of code that work (and in their place, work well).

However, the last two ANSI updates have had "no changes or enhancements" to the language and its roots lie in batch- and single-tasking environments. Likewise, languages such as C, C++, and/or its various clones happen to be favorites of mine if (and only if) I wish to interact on a purely system-level or alien device interface level or if I wish to be a "BITty" purest (nice, but not productive). Writing a user application in C or C++ simply makes no sense as it takes too long, is too susceptible to the "gotcha" principle, and tends to (which really good C programmers would never do) rely too much on supplied C libraries.

The bottom line is finding the right tool (that is, language) to match the need. I present PL/B (its advantages include object-oriented programming, transportability, file access, string manipulation, screen I/O, and program maintenance) as a language tool for your consideration.

Object-oriented programming. The full range of object-oriented, event-driven objects can be implemented on OS/2, Macintosh, Windows, and X-windows. Not only are the traditional GUI objects available (buttons, dialogs, check boxes, radio buttons, drag-and-drop, and so on), but the multimedia aspects of sound, images, and movies are also supported. The developers of PL/B have always worked in true 32-bit environments, so the compilers take natural advantage of this strength of OS/2.

Programmers writing object-oriented applications can use third-party APIs. One vendor offers a WINAPI object-oriented application programming interface. PL/B can act as an object-oriented programming language and still handle the raw ASCII data still used in most businesses. PL/B developers approach object-oriented programming with the awareness that sometimes we must revert to data, not object, processing.

Programmers entering the object-oriented programming world are often confused by the concepts of pixel-based presentation. Many developers let programmers switch between pixel-based references or column/row references (PL/B performs an internal conversion).

Transportability. PL/B is most often a combination of a compiled and interpretative language and is inherently multiuser- and multitasking-oriented (if allowed by the operating system). In fact, PL/B's earliest roots, even on the old Datapoint equipment, were multiuser- and multitasking-oriented.

PL/B source code need never be converted for use on different machines or operating systems. Rather, it can be enhanced to take advantage of performance characteristics of whatever machine and/or operating system is being used. Such enhancements do not affect performance on other systems.

PL/B is easy to learn even by non-technical users because of its English-like structure and easily documented source code. The rule-of-thumb for PL/B programmers is to think of what you want to do, imagine a name you might call the verb, and check the manual to see if that is what it is called. This is also true for object-oriented programming with verbs such as STATTEXT, PICTR, MOVIE, BUTTON, CHECKBOX, RADIOGRP, and so on.

Only the compiled code need be supplied to users. This will prohibit tampering with the purpose of a program, as well as protect the proprietary work of programmers.

Even programs designed and written in a text-based environment can be ported to, compiled in, and run in object-oriented environments. Also, true objects have the appearance and control functions necessary and can be included and controlled via compile time directives in what was once text-based programs to take full advantage of such environments. If PL/B does not currently support the desired object-oriented function, one vendor has introduced a verb known as WINAPI, so even third-party APIs can be integrated into the code.

File-accessing methods. Most other programming languages that employ database-searching mechanisms must use specially formatted files that contain "field attributes" in order to perform their searches. In effect, this corrupts the integrity of the data and makes the data specific to only one language or search method. All four of PL/B's accessing methods maintain the integrity and never corrupt the characteristics of the ASCII and/or native file structure of the operating system. In this manner, the programmer can choose the fastest or most flexible access method for a particular section within the program. The only exceptions to this would be if programmers wanted to do something unusual (because PL/B can write direct HEX, OCTAL, or even Binary data) or if they were accessing an external database through SQL or ODBC interfaces.

The performance characteristics of the most commonly used PL/B types

of file access methods are most impressive. They also cause very little disk read/write head movement.

PL/B supports a database-like accessing method known as AIMDEX that actually gets faster if given more and/or varied information to search a file for (keys). Please note the use of the phrase "database-like" because it operates unlike most database systems and does so on any file structure as well as RPG-formatted files.

Within the confines of a single program, a single data file can be accessed by all four accessing methods in one or more ways. Also, a single data file may be accessed in multiple ways using multiple accessing methods. Even "old" programming languages can still access this type of data.

A standard query language (SQL) interface has been implemented by some of the developers to access traditional databases. Other vendors are approaching this concept of open systems via open database connective (ODBC) that lets any machine access any other machine and/or machines acting as servers.

String manipulation. PL/B has very strong string manipulation characteris-

tics (that is, massage a line or field of data) that, according to many students of PASCAL, exceeds PASCAL's string-manipulation strengths. No one yet has challenged PL/B string-manipulation ease of use or strength. A full compliment of verbs exist just to let programmers logically analyze the contents of a string of characters from whatever the source (that is, objects, disk, keyboard, port, or even from the screen itself).

Large records can be read in a single-disk read, thus obtaining data with minimal disk-head movement. In fact, most versions support data and record lengths limited only by the size of a disk. Even doing partial I/O is allowed if the application requires the program to "walk through" the data on the disk in increments as small as one byte at a time.

The string manipulation instructions are very phonetic and easily learned and let string data be viewed logically, not just physically. The concept of a logical string may be new to many users, but in simple terms, it could be analogous to considering a word or even a single character within a sentence. The program can search,

scan, and/or parse this sentence into its various components.

Interactive screen I/O. Windowing, screen saving, up-down-left-right scrolling, line or character insertion, graphics, color, and many more control features are implemented in many of the PL/B compilers. If a program has these features but the hardware does not support them, they are simply ignored, and the program still runs. Three-dimensional images, moving pictures, and sound are available for GUI environments with more multimedia functions under development.

Extensive control and data-checking features let programmers verify the type, size, and structure of the data entered by users. Most screen and printer controls begin with an asterisks (*) and are referred to as list controls.

System/program maintenance. PL/B is not a wordy language. Simple English statements, as shown earlier, will do what requires extensive coding in many other languages.

PL/B object code actually compiles to be smaller than the source code. Thus, a program can do more without having to be splintered into many other programs. This has a sig-

Visual Basic
PL/B

The Award Winning, Multi-Function Enhancements for OS/2[®]

NEW
VERSION 1.51
WITH
Warp Support!

DeskMan/2[™]

DeskMan/2[™] Workplace Shell[™] Manager

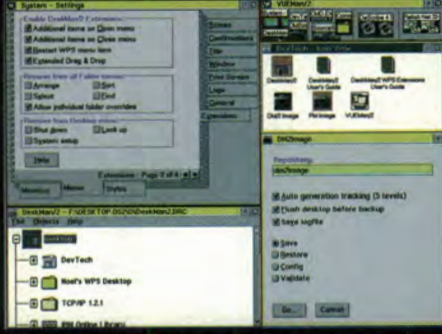
Efficiently Manage Your Workplace Shell!

- Easy to use drag & drop interface
- Complete command line interface + REXX & CID support
- Manage standardized or personalized desktops, customizing access to features
- Selectively save and recreate objects; even transport them between desktops, computers & versions of OS/2
- Recover from or prevent accidental changes to your desktop
- Much more - total management of the Workplace Shell

VUEMan/2[™] Virtual Desktop Manager

Increase Your Productivity!

- Makes multitasking easy; use and organize many concurrent activities with an easy to use drag & drop, point & click user interface
- Switch effortlessly between as many as 81 "virtual desktops" - like having 81 monitors that fit right on your desk!
- Drag and drop objects from WPS to virtual desktops, and between virtual desktops
- "Layouts" record window position and size for automatic repositioning and re-sizing at any future time
- Password protection for windows



All this functionality & much more Four Great Solutions - One Low Price!

"DeskMan/2 dramatically improves the ability of corporations and users to get the most out of OS/2. DeskMan/2 helps administrators restrict and control what users are doing and provides robust backup/restore features and seamlessly enhances the features of the WPS."
Dr. Mike Kogan:
Lead Architect of OS/2 2.0
Author: The Design of OS/2

DM2Image[™] Configuration Snapshot Facility

Saves Your Working Environment!

- Save or restore an unlimited number of complete system configurations (including active or inactive WPS desktops)
- Terrific for disaster recovery; training & demo machines: other uses
- Extensible rules let you customize configuration definitions
- Configuration snapshots are compressed for maximum efficiency
- Protects WPS desktop, config.sys, startup.cmd, .INI files, autoexec.bat, WIN-OS/2 desktop & system config. data

DeskMan/2 Extensions Workplace Shell Extensions

Configure the Workplace Shell to Work Your Way!

- Modify and enhance the runtime behavior of the Workplace Shell
- Enjoy drag & drop without using special keys to shadow or copy
- Control, selectively or globally, all WPS defined object styles, menu items and capabilities - the ability to Arrange, Sort, Delete, Rename, Settings, Lockup, Shutdown, etc. (even object visibility!) is completely definable
- Corporate Edition available with password protection for objects, audit trailing for WPS activities, and other custom enhancements

Copyright 1994 Development Technologies, Inc. All rights reserved. DeskMan, DeskMan/2, VUEMan, VUEMan/2 and DM2Image are trademarks of Development Technologies, Inc. CompuServe is a registered trademark of CompuServe, Inc. Workplace Shell is a trademark of International Business Machines Corporation. IBM, OS/2 and TalkLink are registered trademarks of International Business Machines Corporation.

Available and Ready for OS/2

Available and Ready for IBM LAN

EDITORS CHOICE AWARD OS/2 MAGAZINE 1994

Best of OS/2 1994 OS/2 Inside Reader's Choice Award

DevTech
Development Technologies, Inc.
Software Development & Technology Transfer
308 Springfield Road, Forest Acres, SC 29206-2113
Voice: (803) 790-9230 Fax: (803) 738-0218

Circle Reader Service Number 9

JULY/AUGUST 1995

Connect to the Future of Desktop Computing

Attend the SHARE Technical Conference and discover the powerful connections that exist in the area of desktop computing. SHARE offers more than 100 user sessions, tutorials, concept overviews, vendor demonstrations and hands-on labs on current industry topics like:

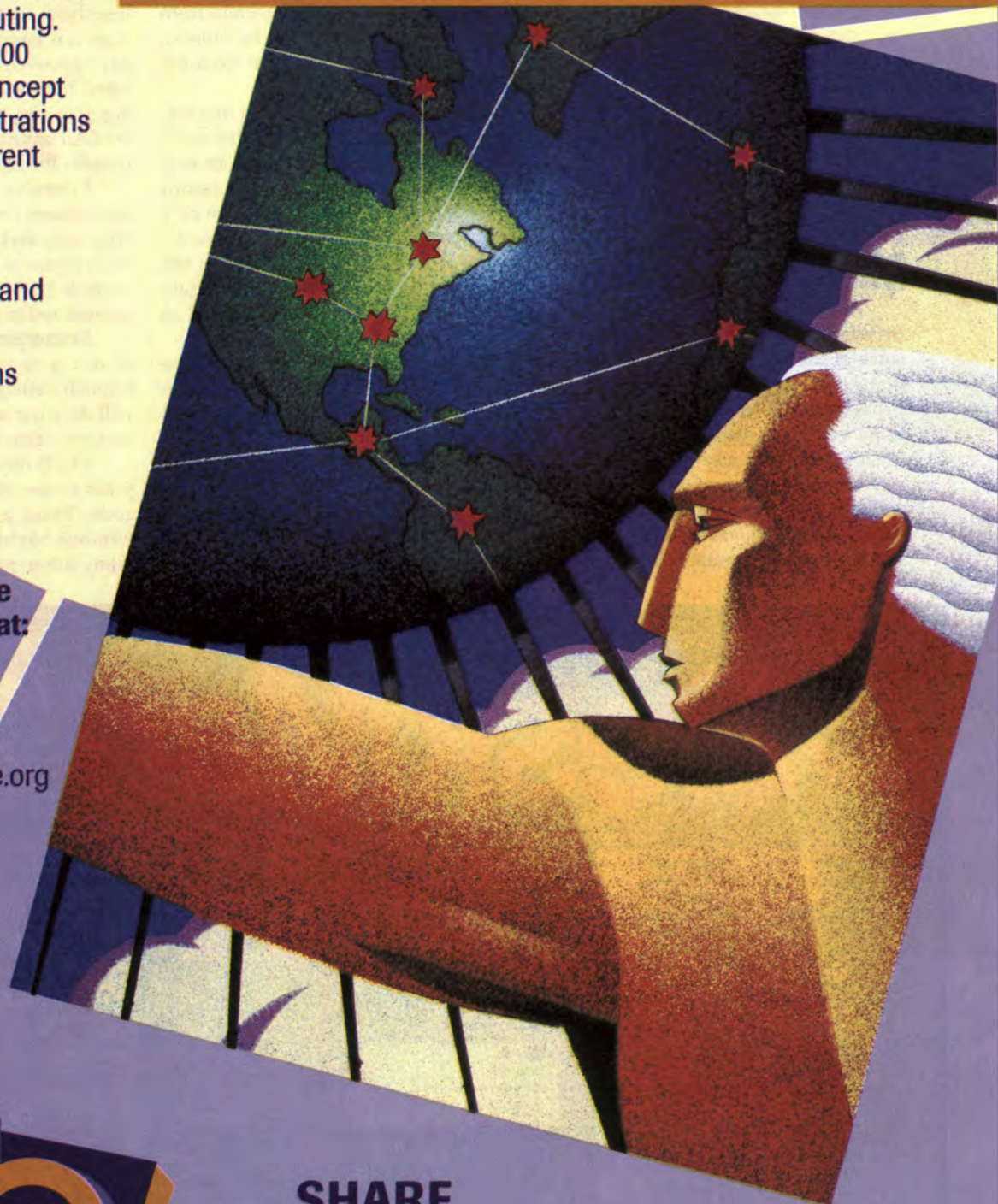
- OS/2 Warp
- Desktop Connectivity and Local Area Networks
- Multimedia Innovations
- Lotus Applications
- REXX
- Desktop Applications

For complete conference details, contact SHARE at:

Phone: 312/822-0932

Fax: 312/644-6363

Internet: sharehq@share.org



**SHARE
Technical Conference
August 13-18, 1995
Orlando**

nificant impact on any computer by reducing overhead to the operating system. It obviously also saves disk space. In OS/2, modules can be loaded so that multiple tasks can share ("thread") the same code in memory, thus reducing memory thrashing and improving performance.

A large PL/B program can be built in modules and each module tested independent of the overall program. Thus, the development and maintenance of an applications system is enhanced as well as simplified. PL/B has been described as being like a Leggo construction block toy; once the pieces are built, you can reuse them over and over in building other programs.

The error-detection characteristics of PL/B allow the development of programs that, in many cases, literally can be self-repairing for damaged data files. In fact, only one possible error condition in the entire language cannot be "TRAPped" for, and that is one caused only by a programmer's error in writing the program. Furthermore, even if a programmer fails to TRAP for an error, PL/B always returns to the originating program "with the error code intact" (rather than having a complete failure of a system of programs), so that error can be written to disk or at least displayed to an operator. Programmers can also TRAP for any character (ASCII or non-ASCII) as well as the full range of keyboard characters (UP, DOWN, HOME, ESC, and so on).

As would be true of any multitasking programming language, complete background and foreground tasking is afforded. Also, the vendors supply a number of separate utilities (for example, INDEX, AIMDEX, an extended SORT utility, file REFORMATING, and so on.) that can also be executed either as foreground, background, shell, or batch programs either independently or directly from the PL/B program.

In many versions of PL/B, direct

communication using industry-standard protocols is available. PL/B is supported on local area networks (LANs) and multitasking environments such as OS/2. However, it works just as well in a single-user environment. No changes need be made to the PL/B program. Several program generator systems are available to aid in the quick production of code in a business environment.

SUMMARY

PL/B is easily learned yet has the capability to handle the most complex programming needs. PL/B is not so wordy nor inflexible as to allow for no margin of human frailty and notes when you make a mistake. If PL/B doesn't have the capability you may need for a special application, it can incorporate your instructions written in assembly or C. You could literally make up your own instructions, although experience has proven this to be an extremely rare need.

PL/B, already very rich in its substance, is still growing. Most will acknowledge that C and C++ are very strong languages, but few would consider them high-level programming languages. C and C++ are still much closer to assembly than to COBOL or BASIC.

PL/B offers the COBOL, BASIC, RPG, dBASE, and such programmers a logical and powerful avenue into the world of object-oriented programming, including full GUI support. Even Visual Basic programmers have found PL/B to be easier to learn, more productive, and (of course) not vendor specific. Therefore, PL/B may well be the applications programming language of the 1990s and beyond.

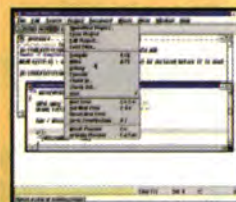
Bill Blocker is systems coordinator and chief instructor at Computer Know-Le-Edge, which is an educational facility, not a reseller of PL/B compilers. For additional information on PL/B you may contact Bill at (713) 784-1762 or bblocker@aol.com.

PL/B COMPILER DEVELOPERS

Sunbelt Computer Systems.....(903) 561-6005
Subject/Will Computer Systems.....(708) 572-0240
Infopro, Inc.(708) 571-4636
Netsoft, AB45-290-21910 (Sweden)
Keech Computer Software(705) 495-1232 (Canada)
Transportable Software(708) 916-7878



OS/2, Win95, WINDOWS, WINDOWS NT



**\$199
SPECIAL
FOR
OS/2!**

Award Winning Visual SlickEdit is now the standard in programmer's editors. You can completely customize it to suit your programming needs. Program more efficiently using extensive language support, project management, syntax color-coding, and SmartPaste™. Save time using comprehensive CUA, Brief, Emacs, and VI emulations.

Powerful Features:

- * Object oriented C-style macro language and DLL interface
- * Incremental search/search & replace
- * Drag & drop editing
- * Line, column, character selection
- * Spell check
- * comments & strings
- * Clipboard Inheritance™ (patent pending)
- * Undo/redo to 32,767 steps
- * Interactive file compare
- * Programmable file manager
- * Macro recording

Cut your work in half using:

- * Procedure tagging
- * Syntax color-coding, expansion, and indenting
- * SmartPaste reindents pasted or dropped source code according to nesting level
- * Compiler error processing
- * Entire manual on-line
- * Multiple clipboards
- * CUA, Brief, Emacs, and VI emulations
- * Built in dialog editor
- * Configurable menus and buttons

To order, Call 800-934-EDIT

Also (919) 831-0600

FAX (919) 831-0101

Compuserve 72662,3312

MicroEdge also makes SlickEdit available for DOS and 80 UNIX platform. Visual SlickEdit supports C/C++, Pascal, FORTRAN, Basic, dBASE, Modula-2, Assembly, COBOL, and Ada. The Windows NT version is available for Intel, Mips, Alpha AXP, and Power PC machines.



MicroEdge PO Box 18038, Raleigh, NC 27619

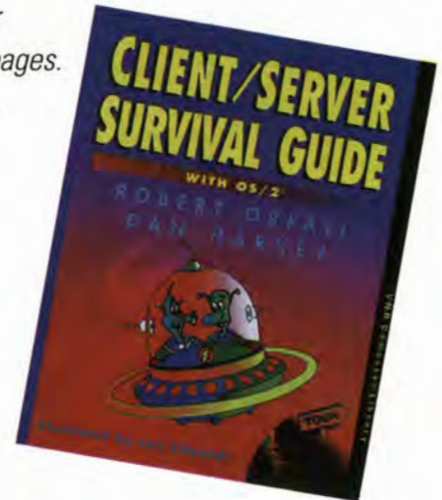
Circle Reader Service Number 10



Object-Oriented Programming

This article is excerpted with permission from *Client/Server Survival Guide with OS/2* (John Wiley & Sons, 1994). 976 pages. ISBN 0-471-13118-0. IBM Publication Number SR28-5494.

By **ROBERT ORFALI** and **DAN HARKEY**



The Server Side of CORBA



Robert Orfali



Dan Harkey

In "Client/Server Programming with CORBA Objects" (September 1994, p. 46), we gave a brief introduction to client/server programming CORBA style. In this article, we take a closer look at what happens on the server side of a CORBA Object Request Broker (ORB). First we go over some of the new features in CORBA 2.0. Then we look at the magic that object adapters provide on the server side of the ORB. We conclude with an overview of how Distributed SOM (DSOM) implements some of these server features.

THE CORBA 2.0 OBJECT REQUEST BROKER

Figure 1 shows the client and server sides of a CORBA ORB. The light areas are new to the CORBA 2.0 specification adopted in December 1994. Even though there are many boxes, it's not as complicated as it appears. The key is to understand that CORBA, like SQL, provides both static and dynamic interfaces to its services.

Let's first go over what CORBA does on the client side:

- The **client IDL stubs** provide the static interfaces to object services. These precompiled stubs define how clients invoke corresponding services on the servers. The services are defined using IDL, and both

client and server stubs are generated by the IDL compiler.

- The **Dynamic Invocation Interface (DII)** APIs let you discover the method to be invoked at run time. CORBA defines standard APIs for obtaining the service definitions, generating the parameters, issuing the remote call, and getting back the results.
- The **Interface Repository** APIs allow you to obtain and modify the descriptions of all the registered component interfaces, the methods they support, and the parameters they require. CORBA calls these descriptions method signatures. The Interface Repository is a run-time database that stores, updates, and manages object interface definitions. The APIs let your programs access and update this information.

With CORBA 2.0, ORBs provide global identifiers, called Repository Ids, to uniquely and globally identify a component and its interface across multivendor ORBs and repositories. The Repository Ids are system-generated unique strings that are used to maintain consistency in the naming conventions used across repositories—no name collisions are allowed. Repository Ids are generated via pragmas in the IDL; the pragmas specify whether to generate them via

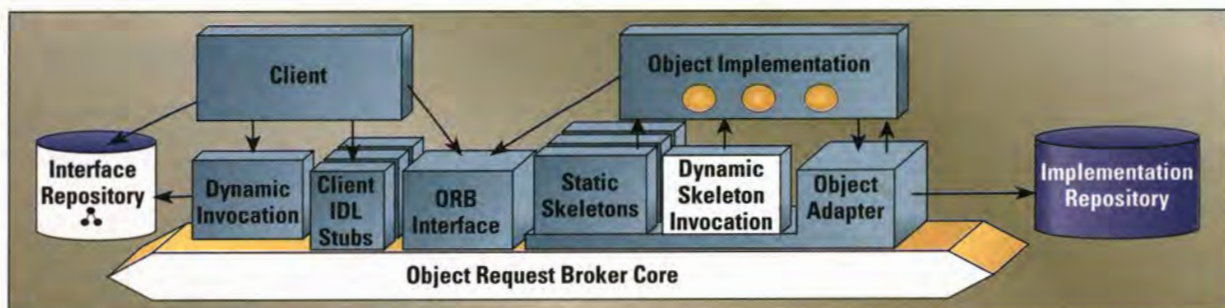


Figure 1. The structure of a CORBA 2.0 ORB.



DCE UUIDs or via a user-supplied unique prefix appended to IDL-scoped names. The Repository Id itself is a string consisting of a three-level name hierarchy.

- The **ORB Interface** consists of a few APIs to local services that may be of interest to an application. For example, CORBA provides APIs to convert an object reference to a string and vice versa. These calls can be useful if you need to store and communicate object references.

The support for both static and dynamic client/server invocations, as well as the Interface Repository, gives CORBA a leg up over competing middleware. Static invocations are easier to program, faster, and self-documenting. Dynamic invocations provide maximum flexibility but are difficult to program; they are very useful for tools that discover services at run time.

The server side cannot tell the difference between a static or dynamic invocation since they have the same message semantics. In both cases, the ORB locates a server object adapter, transmits the parameters, and transfers control to the object implementation through the server IDL stub (or skeleton). Here's what CORBA elements do on the server side of Figure 1:

- The **server IDL stubs**, called skeletons, provide static interfaces to each service exported by the server. These stubs, like the ones on the client, are created using an IDL compiler.
- The **Dynamic Skeleton Interface (DSI)**, introduced in CORBA 2.0, provides a run-time binding mechanism for servers that need to handle incoming method calls on behalf of components that do not have IDL-based compiled skeletons (that is, stubs). The Dynamic Skeleton looks at parameter values in an incoming message to figure out who it's for—that is, the target object and method. In contrast, normal compiled skeletons are defined for a particular object class and expect a method implementation for each IDL-defined method. Dynamic skeletons are very useful for implementing generic bridges between ORBs. They can also be used by interpreters and scripting languages to dynamically generate object implementations. The DSI is the server equivalent of a DII. It can receive either static or dynamic client invocations.
- The **object adapter** sits on top of the ORB's core communication services and accepts requests for service on behalf of the server's objects. It provides the run-

time environment for managing server objects. We discuss object adapters at length later.

- The **Implementation Repository** provides run-time directory information about the classes a server supports, the objects that are instantiated, and their IDs. It also serves as a common place to store additional information associated with the implementation of ORBs. Examples include trace information, audit trails, security, and other administrative data.
- The **ORB Interface** consists of a few APIs to local services that are identical to those provided on the client side.

CORBA SERVERS

What does an object implementation need from an ORB on the server side? It needs a server infrastructure that registers the application's classes, instantiates new objects, gives them unique IDs, advertises their existence, invokes their methods when clients request it, and manages concurrent requests for their services. If we want to get more sophisticated, we could add transaction management, load balancing, and fine-grained security to the list. In other words, we need a program that takes raw class libraries and transforms them into a multiuser server environment. We're talking about the equivalent of a TP Monitor for objects. So who does this type of work? The Object Adapter.

WHAT'S AN OBJECT ADAPTER?

The Object Adapter is the primary mechanism for an object implementation to access ORB services (Figure 2). It provides a total environment for running the server application. Following are some of the services provided by the Object Adapter:

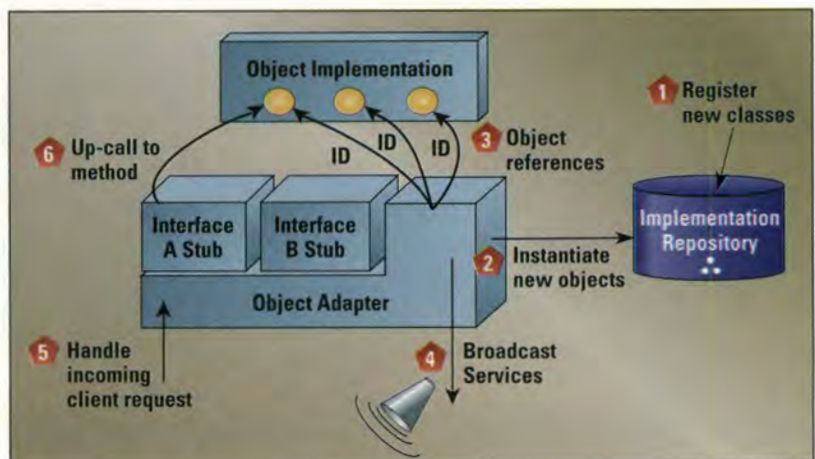


Figure 2. The structure of a typical object adapter.

1. *Registers server classes with the Implementation Repository.* You can think of the Implementation Repository as a persistent store that is managed by the Object Adapter. Object Implementation classes are registered and stored in the Implementation Repository.

2. *Instantiates new objects at run time.* The Object Adapter is responsible for creating object instances from the implementation classes. The number of instances created is a function of the incoming client traffic loads. The Adapter is responsible for balancing the supply of objects with the incoming client demands.

3. *Generates and manages object references.* The Object Adapter assigns references (unique IDs) to the new objects it creates. It's responsible for mapping between the implementation-specific and ORB-specific representations of object references.

4. *Broadcasts the presence of the object servers.* The Object Adapter may broadcast the services it provides on

the ORB or it may respond to directory-type queries from the ORB core. It is in charge of letting the outside world know of the services it manages. Eventually, we expect to see some tight levels of integration with global directory services such as X.500.

5. *Handles incoming client calls.* The Object Adapter interacts with the top layer—typically an RPC—of the ORB core communication stack, peels off the request, and hands it to the interface stub. The stub is responsible for interpreting the incoming parameters and presenting them in a form that's acceptable to the object's method invocation.

6. *Routes the up-call to the appropriate method.* The Object Adapter is implicitly involved in the invocation of the methods described in the stubs (or the skeleton). For example, the Object Adapter may be involved in activating the implementation, and it can authenticate the incoming requests.

An Object Adapter defines how an object is activated. This can be done through the creation of a new process,

the creation of a new thread within an existing process, or the reuse of an existing thread or process.

DISTRIBUTED SOM (DSOM)

The rest of this article looks at how DSOM implements these CORBA features. DSOM is a CORBA-compliant ORB from IBM that allows SOM objects to talk to other SOM objects across address spaces on the same machine or across networks. SOM insulates clients from the object implementation; DSOM adds to that object location insulation. As we go to press, DSOM is the most comprehensive implementation of OMG's CORBA standard for ORBs. It currently allows OS/2 objects to interoperate with objects on AIX and DOS/Windows. Apple is implementing DSOM on the Macintosh as part of OpenDoc, and DSOM implementations are expected within a year on OS/400, MVS, and various UNIX platforms. Eventually, DSOM will interoperate with every ORB via CORBA 2.0 bridges.

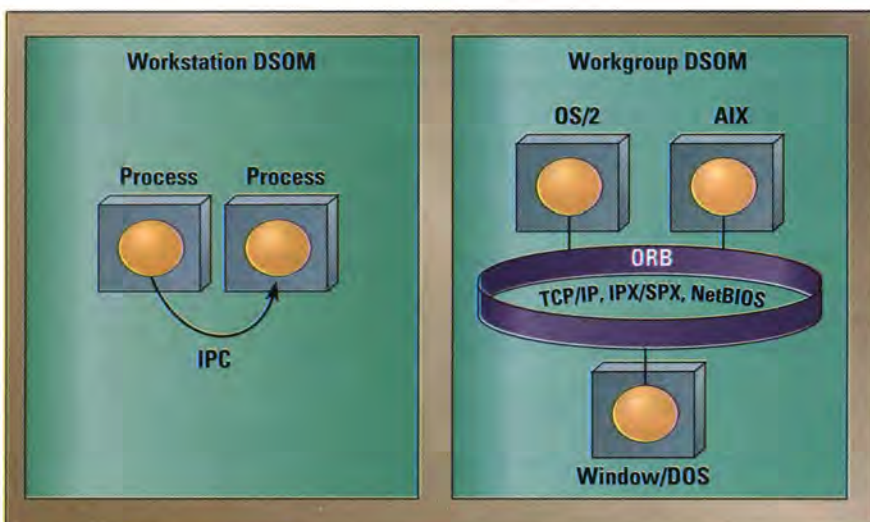


Figure 3. DSOM: Workstation vs. Workgroup.

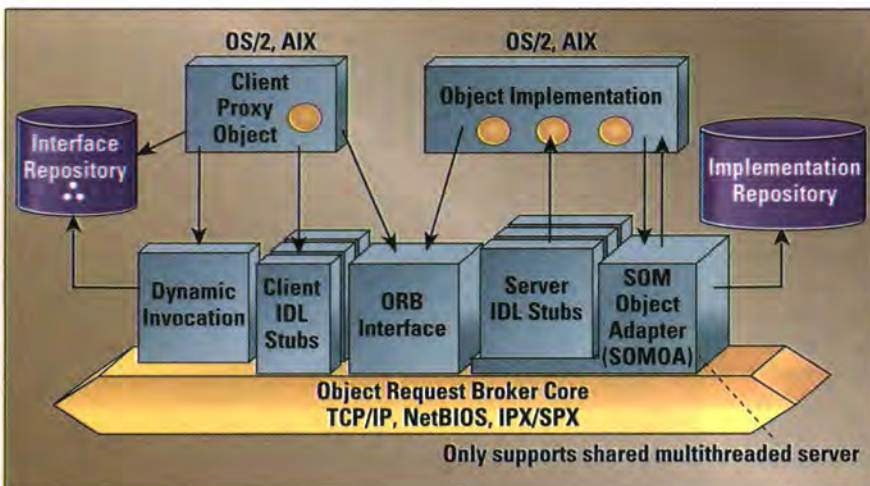


Figure 4. DSOM is a CORBA 1.1-compliant ORB.

DSOM: WORKSTATION VS. WORKGROUP

DSOM is simply distributed SOM. It uses the standard SOM IDL compiler, SOM APIs, language bindings, and dynamic object model. Workstation DSOM supports client/server object interactions among processes on the same machine; it uses existing interprocess communication facilities (IPCs). Workgroup DSOM supports client/server object exchanges across networks of OS/2, AIX, and Windows machines. It uses socket-based TCP/IP, IPX/SPX, and NetBIOS stacks, and it can be customized to other stacks. DSOM programs run unchanged in both environments (Figure 3); they use the same language bindings and stubs. DSOM is packaged as a client and server DLL addition to SOM.

THE DSOM ORB

Like all good ORBs, DSOM makes its living by providing interobject client/server communications (Figure 4). This includes creating, destroying, identifying, locating, and invoking methods on remote objects. It also includes managing object references. The default Basic Object Adapter that ships with DSOM—called SOM Object Adapter (SOMOA)—supports only the CORBA shared server activation policy. SOMOA's multithreaded implementation supports multiple objects within the same server process.

DSOM supports both the CORBA Interface and Implementation Re-

positories. The Interface Repository describes the behavior of objects; the Implementation Repository deals with their location and packaging. DSOM provides an open architecture that lets you subclass and modify the classes that provide the proxy client, server, and socket functions. As a result, you can tailor the key pieces of the ORB implementation to suit your needs. Of course, it takes some serious programming to do that.

DSOM IMPLEMENTATION DETAILS

The DSOM architecture is very much that of a traditional CORBA ORB with a shared server BOA. This section goes into the next level of detail of how the DSOM ORB is implemented in the OS/2 environment. In Figure 5, we walk through a scenario that includes starting DSOM on both the client and server and invoking a method on a car object. We then watch its execution on the server and what happens at each step:

1. *Set environment variables.* The DSOM clients and servers must set environment variables that give the pathnames of the directories (local or remote). These directories contain the implementation and interface repositories

and the location of the class DLLs.

2. *Register the class implementations.* **Regimpl** is an interactive tool that lets you query and update the contents of the Implementation Repository. Use it to define the server's program name, the class implementations it supports, and the location of the server machine.

3. *Start the DSOM daemon, SOMDServer, and SOMOA.* Issue **start somdd** on the server to start the SOM daemon, which starts the default server program object of class **SOMDServer** and a **SOMOA** object adapter. The program registers the classes it implements

and the active object IDs in the Implementation Repository. It then notifies SOMOA that it is ready to accept requests by invoking the **impl_is_ready** method. Each DSOM server must contain an Object Adapter of class **SOMOA** (Figure 6). The **SOMOA** object registers the socket port with the DSOM location daemon, handles communications with the ORB, executes the main server loop, dispatches the next method request to the **SOMDServer** object (Figure 7), and keeps running until the **SOMDServer** program invokes the **deactivate_impl** method.

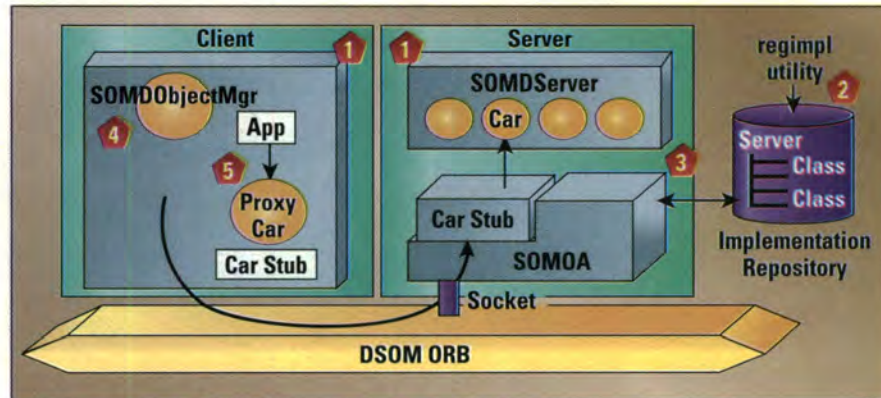


Figure 5. A detailed DSOM scenario.

"THERE JUST AREN'T ANY OS/2® APPLICATIONS" (YEAH, AND THE EARTH IS FLAT)

dbfREXX and dbfLIB

by dSoft Development

New! Version 2

Includes Support For NDX & NTX Files!

dbfREXX:

This dynamic link library provides simple, affordable database management from your OS/2 REXX programs. Now accessing dBase files is unimaginably easy.

DSF25 MSRP \$99.00 IB \$95.00

dbfLIB:

The solution for harried MIS developers! Allows multiple users to access dBase files. Provides a consistent set of API's across OS/2, DOS, Windows & Windows NT.

DSF22 MSRP \$195.00 IB \$159.00

VisualAge



by IBM

IBM's powerful new vision of programming! VisualAge is a client/server application development power tool that focuses on line-of-business applications. Components include visual programming tool, parts library, GUI support, client/server & communication support, enhanced DLL support, multimedia support and much, much more!

17H7495 Single User V2

MSRP \$2,495 IB: \$1,550

17H7496 Team V2 OS/2

MSRP 4,995 IB: \$3,100

THESE ARE A FEW OF THE
HUNDREDS OF OS/2
SOLUTIONS AVAILABLE
RIGHT NOW FROM
INDELIBLE BLUE, INC.

CORPORATE DISCOUNTS
AVAILABLE THROUGH THE
VALU PROGRAM

CALL FOR NEW FULL
COLOR CATALOG!



INDELIBLE
BLUE
Inc.

THE SINGLE SOURCE
FOR OS/2® SOLUTIONS

OS/2, VisualAge and IBM are registered trademarks of IBM Corporation. All other trademarks belong to their respective owners. Prices are subject to change.

ORDERS: 1-800-776-8284

Circle Reader Service Number 11

4. Start the client Object Manager. This is an object of class SOMDObjectMgr that knows how to find objects on the network (Figure 8). For example, it can find a server that implements a particular class of objects or a remote

object given its ID. It also provides Lifecycle services for proxy object creation and destruction on the client workstation.

5. Invoke the method on the client proxy. Remote server objects are repre-

sented in the client by local proxy objects. Methods invoked on the local proxy are transparently redirected to the remote object. The proxy object inherits the interface of its target object; each method call is automatically overridden with a remote dispatch call. The dispatching logic is inherited from SOMDObject and SOMDClientProxy (Figure 9). The proxy object call cooperates with the local Object Manager to find a server, marshals method arguments, converts object pointers in the clients, sends the method to the server, and later receives the method results.

6. Invoke the method on the server. SOMOA receives the request, demarshals the arguments with the help of the IDL stub, authenticates the message, calls SOMDServer to convert the target object ID to a SOM object pointer (SOMDServer loads class DLLs and creates/locates objects as needed), executes the method by invoking the somdDispatchMethod on SOMDServer, marshals the method's return values to the client, and sends the response back to the client.

It should be apparent by now that the entire DSOM ORB is built using objects. This type of design creates well-defined interfaces between the components; the system can be extended and modified at will. The system objects blend with the customer objects, and they feed on each other through inheritance webs. We're getting our first glimpses at the client/server system "objectware components" of the future.

Robert Orfali and Dan Harkey are authors of the award winning and best selling books, *Client/Server Survival Guide with OS/2* (John Wiley & Sons, 1994) and *Client/Server Programming with OS/2* (John Wiley & Sons, 1993). Bob and Dan have been developing client/server applications and tools for the last nine years. They currently work on the application of distributed object technology. Bob and Dan are affiliated with IBM Austin, Texas. They work from the San Francisco Bay Area.

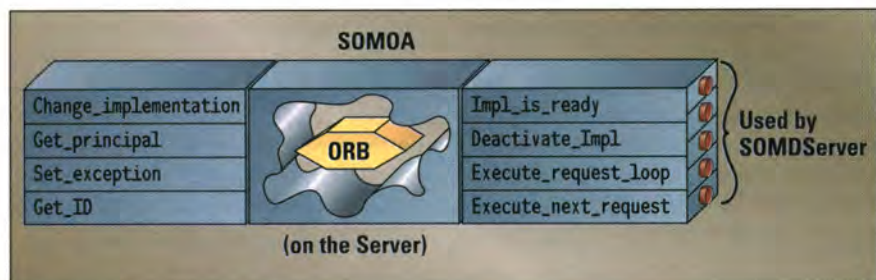


Figure 6. The SOMOA Object Adapter class (on the server).

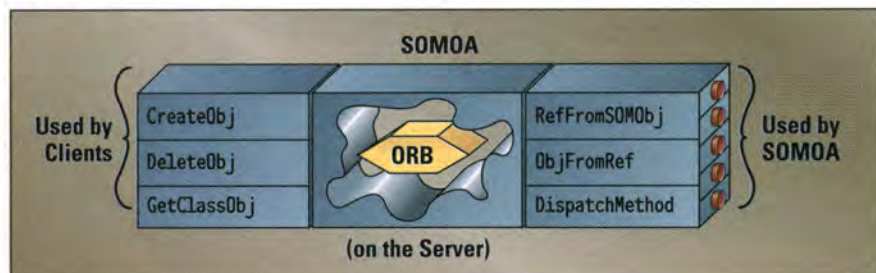


Figure 7. The SOMDServer class (on the server).

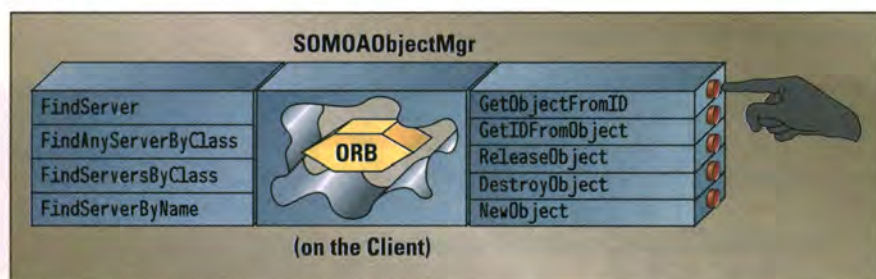


Figure 8. The SOMDObjectMgr class (on the client).

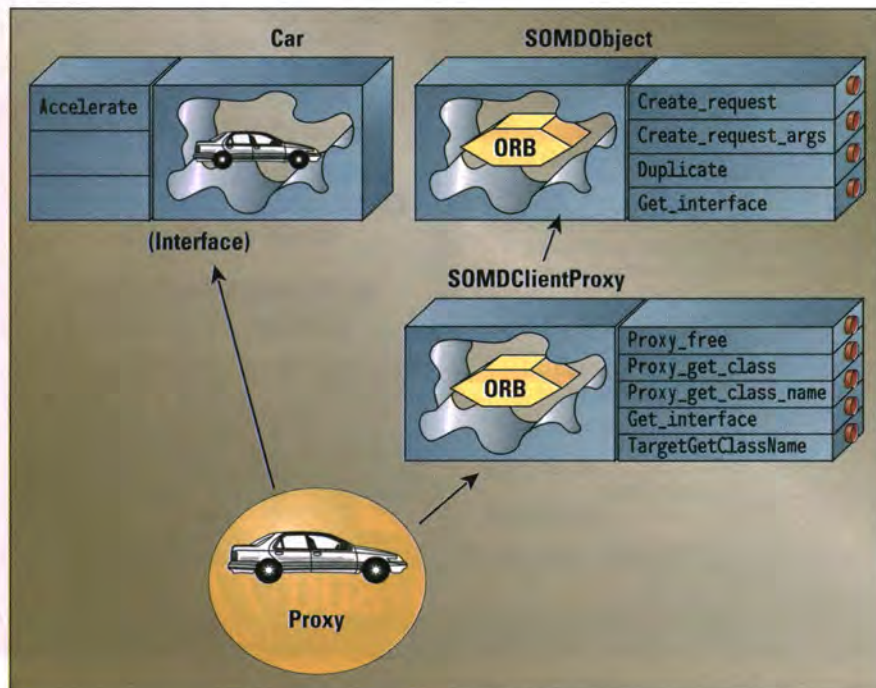


Figure 9. Proxy objects: "Mixin" of Car interface and SOMDObject dispatcher.

REFERENCES

1. Robert Orfali and Dan Harkey, *Client/Server Survival Guide with OS/2*, John Wiley & Sons, 1994, IBM publication number SR28-5494. Part 7 provides a 140-page introduction to distributed objects.
2. Robert Orfali and Dan Harkey, "Client/Server with Distributed Objects," *BYTE* (April 1995).



The delivery of enhancement software is becoming more complex for vendors since business solutions require flexible platforms, short development cycles, and portability. Here are the challenges we faced in our search for a cross-platform development tool.

By MAGGIE SANDERS

Searching for a Cross-Platform Development Tool

We were looking for a cross-platform, client/server development environment in which to build today's products and provide a framework for tomorrow's. Our search led us to an object-oriented mentor and an OS/2 software tool.

Our company, Check Solutions, develops software for the check and image industry and assists banks with the integration of IBM's Check Processing Control System (CPCS), IBM's ImagePlus High Performance Transaction System (HPTS), and Check Solutions' own line of enhancement software. We provide installation services, software, tools, and maintenance support services for financial institutions. Our customers want to shorten the "time to market" for their own product offerings by leveraging our technology and products.

WHAT WAS THE PROBLEM WE WERE TRYING TO SOLVE?

As a vendor, we faced the difficulty of providing a standard solution for building business software that solves today's problems while providing a framework for the future. The delivery of enhancement software is becoming increasingly more complex since our customers' solutions are increasingly based on multiple computing platforms. These distributed, networked environments, together with time-to-market pressure from sales and marketing, present a complex challenge to anyone involved in software development.

Our challenge was to provide an image correction and payment reconciliation facility for the lockbox line of business using an object-oriented approach. This lockbox application would work in association with CPCS and HPTS to offer a high-speed,

image-based remittance solution. And, of course, we had to do this within a constrained time frame.

THE IMAGE-BASED REMITTANCE SYSTEM PRODUCT

Our Image-Based Remittance System (IBRS) has two major client components. The first is the Image-Based Reject Correction (IBRC) facility, which allows financial institutions to correct the OCR scan lines on a payment coupon image and the MICR-encoded lines on a check image in case of smudges, tears, backward or upside-down items, and so forth. The second component of IBRS is the Document Decision and Disposition (D³) facility, which is primarily a rule-driven workstation application that facilitates the reconciliation of payments due (represented

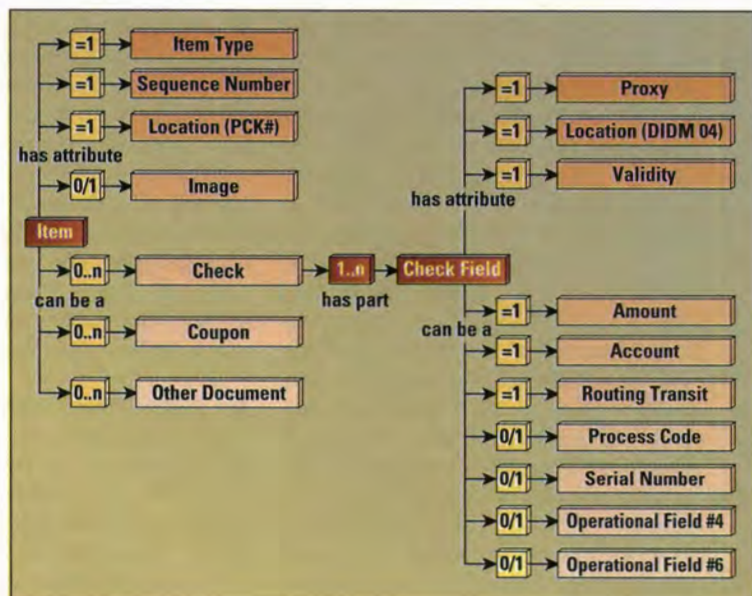


Figure 1. Semantic network.



in the scan line of the payment coupon) with the payment remitted by the consumer (the monetary value of the check). Using subject matter experts as well as our own customers to assist in defining the end product to ensure market success, we began our search for a cross-platform development environment.

ESTABLISHING CRITERIA FOR DEVELOPMENT ENVIRONMENT

The development environment had to be:

1. Object-oriented. This allows for development of cross-platform applications. Object-oriented design would allow us to model the application domain more effectively. We wanted to position ourselves as platform-independent and not tied strictly to a single check/image processing solution.
2. Client/server-oriented. Check Solutions has many applications that run on mainframes. We were concerned about the partitioning and scalability of the processes. For busier environments that need more users and higher transaction volume, we needed the ability to scale up, use more powerful hardware, and make workload run on more than one box.
3. Portable. This allows us to provide the same functionality across different platforms with minimal effort.
4. Able to facilitate the reuse of code for future projects.

EVALUATION AND SELECTION OF OBJECT-ORIENTED MENTOR

Having established the criteria for our development environment, we began looking for an object-oriented software mentoring consultant to help us think in an object-oriented manner. We needed someone with a solid reputation in the object-oriented world and one that had experience dealing with real-world applications. Working from a list of object-oriented mentoring groups found in a technical periodical, we interviewed several companies, including Rational Software Corp., Berard Software Engineering, Select OMT, and Skill Dynamics. We chose Berard because it covered the end-to-end process and had a significant amount of experience in object-oriented design. Berard had an impressive client list, offered highly skilled instructors, was accessible, and was competitive in price. It also offered an object analysis and diagramming tool, called BOCS. Berard provided onsite object-oriented training for six Check Solutions people in August 1994.

EVALUATION AND SELECTION OF TOOLS

Using the aforementioned criteria, we evaluated tools from several vendors, including IBM's Visual Age, PowerSoft's PowerBuilder, Template's SNAP, and the Dynasty tool, all targeted toward full environment. We also evaluated object-oriented class libraries such as XVT, Liant, and SQL Objects ++.

Template's SNAP tool had many attributes we liked—a GUI editor, a fourth-generation language, support for most major database players, a strong distributed object model, portability, the ability to interface with procedures written in either C or C++, and a syntax of class definition that is similar to C++. Most important, however, SNAP had an add-on feature called the Workflow Template, which gave us all the server functions we otherwise would have had to write: queue management, work item routing, restart/recovery, data persistence, hot backup, and so on. The Workflow Template would allow us to graphically design our work flow independent of process partitioning. What the SNAP tool did not have at the time, however, was an OS/2 version. We negotiated with Template in September 1994 to port the SNAP product to OS/2 while we began a parallel project development effort using the Windows version of SNAP. The Berard training we had in August proved invaluable during the SNAP training we attended in September. We started developing the Image-Based Remittance System at the end of September.

OBJECT ANALYSIS

After going through Berard's Object-Oriented Method (BOOM) training and Template's SNAP training, we began the task of converting our requirements to an object-oriented representation. To do this, we used Berard's BOCS tool, which facilitated the analysis and diagramming of our object model. The Berard method, with BOCS, makes extensive use of semantic networks. The semantic network shown in Figure 1 represents an item (a check or coupon) and is a subset of our complete object model.

THE DEVELOPMENT PROCESS

Initial development for the IBRS product was done in the Windows environment. Template targeted January 1995 for delivery of the first OS/2 version of SNAP. One of the first things we had to do was to implement our object model. The definition of the classes in SNAP was very straightforward. A portion of

the SNAP code that implements the semantic network is shown in Figure 2.

TWO MAJOR HURDLES TO OVERCOME

While using SNAP in our development effort, several things were missing, from our point of view. Support existed for bitmaps but not for image manipulation in general. Further, SNAP's online help function provided only a simple text window. It did not contain hypertext or graphic capabilities or a table of contents, and a single source document could not be used in creating the help. We would have to develop these missing pieces.

NO SNAP SUPPORT OF IMAGE MANIPULATION

The IBRS product had to be able to decompress an image file that had been compressed in IBM's ABIC format and downloaded from IBM's HPTS system. It then had to be able to display the image so we could perform basic image manipulation, such as zoom, rotate, and scale to fit.

We acquired a decompression library from IBM that would decompress images that came down from HPTS. We evaluated several image libraries that were available commercially and settled on the HALO Imaging Library from Media Cybernetics because of its experience in the marketplace and its full support of both OS/2 and Windows. HALO allowed us to manipulate the decompressed image. To implement the image support function from within SNAP, we wrote 1,500 lines of C code to provide the glue for the HALO Imaging Library and IBM's decompression library. We then created a SNAP class, called Image Window, which encapsulated the image manipulation functionality into a reusable SNAP object that is shared between the IBRS programs.

ONLINE HELP SUPPORT

One of the requirements of the IBRS product was that it support online help. Although the SNAP tool did offer a rudimentary help feature, we wanted hypertext and graphics capabilities, table of contents capabilities, and the ability to write the help in a single-source document, regardless of whether we were in Windows or OS/2. After evaluating many online help tools, we settled on Virtual Media Technology's HDK, a Windows-based help tool that overlays Microsoft Word. SNAP was not CUA-compliant and did not have function key support. Therefore, there was no direct support

for the F1 key. To create the interface for HDK and make help functions available to the IBRS programs, we wrote 600 lines of C code.

To make the help work in SNAP, we created a subclass of SNAP's "TEXT ENTRY" class and of its "MENU" class. We added attributes in the subclasses for help file name and context ID. In C code, we subclassed the SNAP window and translated the F1 key into a character that SNAP could recognize. Since SNAP's default window proc ignores the F1 key and does not pass it up the line, the IBRS programs couldn't see it. By subclassing the F1 key, we translated it into one the SNAP window proc will pass on to the IBRS programs. When our TEXT ENTRY subclass sees the translated character, it kicks off the help system, using the help file name and context tag defined in the subclass.

PORTING TO THE OS/2 VERSION OF SNAP

With the exception of the two C programs for online help support and image manipulation support, all other SNAP programs simply needed to be recompiled using the new OS/2 version of SNAP. Template had done a good job creating the first OS/2 version. There were minor bugs, but for the most part, everything worked well. By mid-February, after continual interchange between Check Solutions and Template, reporting bugs, and testing, we had a solid beta version of SNAP 6.5 for OS/2.

We ported the online help support and image support over to OS/2 ourselves. Three weeks after we received the OS/2 version of SNAP, we were back to the same level of function we had had in Windows. The remaining development was done using the OS/2 version of SNAP.

DEVELOPING THE SERVER COMPONENT

We used SNAP's Workflow Template to develop our server component, which receives files from the host (CPCS) system, then breaks them up into segments of work for distribution to the client. Clients send work back to the server, which sends it back to the host. All routing is controlled by the Workflow Template. It took only a week to get the Workflow Template operational.

SUMMARY

The Image-Based Remittance System was our first excursion into the object-oriented world. It was, for us, a transitional object-oriented project. That is,

during project development, we used as much object-oriented design as we could considering the constraints we were under. Our next project will be object-oriented from inception to completion. Our relationship with Berard Software Engineering is ongoing, with plans for more training.

Despite the constraints we were under—porting the product to the new version of SNAP in mid-development, a learning curve for object-oriented design and the SNAP tool, and the enhancements we had to make to the basic tool to allow for image and Win help support—we brought a major product to market in eight months. We went from idea conception to completion with only three full-time people. We believe our initial goals of tool usage, the design and building of an evolving prototype with little throw-away, reuse potential, and providing a

framework for tomorrow's products were achieved.

Maggie Sanders is the Technical Publications Manager at Check Solutions Co. You can reach Check Solutions at 76336.2173@CompuServe.com.

REFERENCES

Berard Software Engineering, Inc., 902 Wind River Lane, Suite 203, Gaithersburg, Md., 20878. Voice: (301) 417-9884; Fax: (301) 417-0021.

Template Software Inc., 13100 Worldgate Drive, Suite 340, Herndon, Va., 22070-4382. Phone: (703) 318-1200; Fax: (703) 318-3253; info@template.com.

Media Cybernetics, 8484 Georgia Ave., Silver Spring, Md., 20910. Phone: (301) 495-3305; Fax: (301) 495-5964.

Edward V. Berard, "Essays On Object-Oriented Software Engineering."

```

classes:ITEM:
[inherits: WFT WORK ITEM]

attributes:

Schema:                public own member of SCHEMA                [default: unknown].
ItemType:               public sgl (check,coupon)                [default: unknown].
ItemSequenceNumber:    str                                         [default: unknown].
ItemPocket:            str                                         [default: unknown].
ItemFBWFile:           str                                         [default: unknown].
ItemBBWFile:           str                                         [default: unknown].
ItemDocument:          anymember                                   [default: unknown].
ItemDisposition:       member of D3DISPOSITION                    [default: unknown].
ItemStatus:            public sgl(inserted,moved,deleted)         [default: unknown].
ItemUOWFileName:       str                                         [default: unknown].
ItemPrevSeq:           str                                         [default: unknown].

%

classes:CHECK:

attributes:

ListOfCheckFields:      members of CHECK_FIELD                    [default: unknown].

%

classes:CHECK_FIELD:

attributes:

CheckField:            anymember                                   [default: unknown].
FieldNumber:           int                                         [default: unknown].
Location:              member of DIDM04                           [default: unknown].
Validity:              truth                                       [default: unknown].

%

classes:ACCOUNT:

attributes:

Account:               str                                         [default: unknown].

```

Figure 2. Sample code.



Object-Oriented Programming

Object-oriented programming will cause wrenching change to both the technology and business of software. One change may be the advent of a cottage industry, where domain specialists offer their skills in the form of reusable software components.

By RICK RUNYAN

Parts: The Newest Cottage Industry

Object-oriented technology is giving every indication that it will deliver on the promise of reuse. If so, this will be looked back on as the most significant advance in software programming technology since the advent of the compiler some 35 years ago. But just as significant as the technological advance will be the dramatic economic restructuring of the software industry. Real reuse will precipitate a "parts economy" for software. Parts will provide a "make or buy" alternative for developers and their management, and as "make" gives way to "buy," the age of software as a craft will end.

Parts and reuse technology give the software industry the opportunity to apply the discipline of engineering science to the design and construction of software applications. Competitive pressures have been demanding these advances for years. Only now are we seeing the confluence of hardware and software technologies that finally hold the promise of unleashing a response to those pressures in the form of object-oriented systems.

Yet to be generally and unequivocally proven in widespread application and at industrial-strength scale, many enterprise prototype projects are going on, and a generous sprinkling of successful scaled-up, full-production applications are pointing the way. The quantitative benefits from reusing parts are in the process of being proven.

HOW OBJECT-ORIENTED PROGRAMMING ENABLES REUSE

The three essential ingredients to the widespread realization of reuse are development technologies that truly support reuse, libraries of reusable things with ways to classify and find them, and the spread of skills needed to exploit the potential of these systems and parts.

Some will mourn the passing of the era of "software by craft." In a lot of ways, it was fun, especially for those of us who at some time or other donned the mantle of the programming priesthood—perhaps less so for those we were intended to serve. In the end, the true measure of reuse success will be the tendency for programmers to choose buy over make—or, put bluntly, the final defeat of not-invented-here (NIH) attitudes. Applications will be built from preexisting parts.

But all is not lost for those who would craft. After all, who will build the parts? That will remain highly creative, and if not precisely a craft, it will certainly be a more disciplined endeavor. This crafting will be done by two general categories of developers: those who make a business of providing parts for others to use and those who are building applications and do not find the prefabricated parts they need to complete their applications. This latter group will be programming by exception. The former group will be programming by plan. This revised structuring of the creation and building of software will also see the fruits of these efforts be far more open, at least at the black-box interface level, and much more accessible since the industry economics will shift to motivate reuse.

HOW OBJECT-ORIENTED PROGRAMMING ENABLES VISUAL PROGRAMMING

Once introduced, graphical user interfaces (GUIs) and drag-and-drop techniques quickly flourished as a preferable method for manipulating both application navigation and data. The attractiveness of these visual paradigms is rooted in the intuitiveness of object manipulation and the same hand/eye coordination that has helped us navigate our world since childhood.

CONVERGENCE

@VISIONS

Television
that reaches
and **EMPOWERS**.
What good is an
infobahn without the
ability to drive or the
time to learn? ➡

@VISIONS/:)

Convergence of data management,
telecommunications and
multi-media in an entertainment
interface → television.

Humanity + **TECHNOLOGY** + Television =

@VISIONS//:

Empower yourself. Watch and learn

Visions, 25 Seabreeze, Delray Beach, FL 33483
407-279-9333 • Fax 407-279-9309
imageinn@gate.net

visions

motoroladigitalequipmentpardisprogressexideisissilicongraphics
duracellxeroxernst&youaccessgraphicsproximamerisel

Circle Reader Service Number 12



Early GUIs were not implemented in object-oriented languages, however, but rather in procedural languages. Witness Microsoft's Windows right up through Version 3.1 and Macintosh's equally appealing GUI. These systems, and the early applications that emulated them, were written procedurally. That meant that the shift from the apparent object orientation of the GUI to the procedural programming paradigm of the implementing language was done step by step and element by element by GUI programmers. The task was daunting in that it required the thorough understanding of both paradigms and a clear understanding of the bridging that was taking place.

Since it is fairly easy to see that icon manipulation and GUI operation in general are easily mapped to object manipulation, object-oriented systems later proved more appealing to the implementation of such systems. The appeal was that they relieved programmers of the paradigm shift responsibility that was driving them schizophrenic. Written in an object-oriented language, or even a less pure object-based lan-

guage, the responsibility for the paradigm shift was pushed down into the application development system and not left totally to the programmers.

The object paradigm, which so readily facilitates the visual part of an application, can also be applied "behind the glass." This is the simplified distinction between true object-oriented systems and so-called object-based systems.

THE APPEAL OF PARTS TO BOTH PROGRAMMING PARADIGMS

The appeal of reusable parts in both the object-oriented and the visual programming paradigms tends to grow on you. It starts with the seed of an idea and blossoms into full-grown comprehension and proactive cultivation.

With objects understood at the visual level, the appeal of objects to non-visual application components follows. Some more advanced systems offer that abstraction. True object-oriented systems support the conceptualization of an application from the glass right on down through the business logic, the middleware, and into the operating system itself—hence the appeal

and advent of object-oriented operating system shells and operating system functions. OS/2 and Taligent offer prime examples of this trend.

Comprehension of reuse potential begins to grow with understanding of objects both on and behind the glass. The appeal of the object-oriented paradigm shifts from an understanding of its fundamentals and technicalities to an appreciation of its consequences. Reuse is the most obvious. First, there is the notion of reusing objects the programmer developed. We understand best what we have written ourselves. Then there is the recognition that much of the programming in an object-oriented, or even object-based, system is the reuse of system-provided parts. Consequently, there is a great appreciation for application development systems such as VisualAge that are delivered with robust libraries of parts.

Finally, confidence begins to defeat the historical distrust for code written elsewhere, and the prospect of safely reusing commercially available parts becomes not only interesting but also preferred. The value of architected parts interfaces, such as those offered with OpenDoc, becomes high. Confidence finally replaces NIH. The scale swings in favor of buy and away from make. Trust fosters reuse, and in the end, software engineers, like their hardware engineer colleagues, finally stand on the shoulders of their predecessors—instead of their toes.

HOW THE PARTS BIZ GOT ITS START WITH VB BUT IS UNLEASHED WITH OOP

Microsoft's Visual Basic should probably be credited with having the greatest influence on popularizing the notion of the component reuse paradigm for building graphical user interfaces. Smalltalk-based products like IMB's VisualAge, ParcPlace's Visual Works, and Digital's PARTS are more appropriately given credit for extending that paradigm to a more generalized and well-defined component architecture at both the visual and nonvisual levels.

Object orientation facilitates both a solid public interface for components and a visual manipulation programming approach. It is this combination that unleashes the power of today's object-oriented development tools. In the case of VisualAge and PARTS, visual programming has been taken to a new level with function icons being connected with message arrows on the screen. But a visual programming approach to part reuse would not have the acceptance it currently does if

Definitions

Objects. Objects are composed of encapsulated data and the functional code (or methods) that operate on the data and communicate with other objects.

Parts. Parts are the same as objects but also carry the connotation that they are used in a visual builder. Parts are application components with a fixed internal structure and provide services through defined interactions at the part interface generally using an event mechanism. There are "atomic parts" and "compound (or composite) parts." The latter are made up of the former much as compounds in chemistry are made up of atoms. All atomic parts tend to be small, but their atomic "weights" vary somewhat. How weighty they are depends on how much data they encapsulate and how robust the functions are that operate on the data and communicate outside the object.

Compound parts. Compound parts (also known as composite parts), like chemical compounds, can weigh anything, ranging from very small and competing in smallness with some of the atomic parts, to very large. H₂O is an example at one end of the spectrum and DNA at the other. Similarly, there are small compound parts and huge ones.

Class libraries. Collections of atomic parts and small compound parts can be considered class libraries. Classes provide services in a specific domain. Large collections of classes and large compound parts that are designed to work together are considered frameworks. Class libraries tend to be organized to contain related functions and data; frameworks tend to be class libraries with the added value of being organized with some system level structure, often industry specific.

Componentry. While the precise definitions of these terms are still evolving, the purpose here is to note that all can be considered "componentry." Componentry is application function in some state of incompleteness. Components are expected to be put together with other components or customized by an application developer to complete an application for delivery to a specific user.

more fundamental part reuse tools had not blazed the trail. Often things progress in steps, and this is a perfect example of a less robust approach blazing the trail for a more comprehensive approach later on.

The value of these trailblazers was that they offered a first step in "componentizing" for the part of the problem that was easiest to grasp: the visual part. This gave object-oriented and non-object-oriented programmers some practical experience for comprehending the more difficult abstractions supported in fuller object-oriented systems. This allowed a step-wise path through the infamous object-oriented paradigm shift.

One of the drawbacks of "OO only on the glass" development paradigms is that, as mentioned earlier, somewhere the paradigm shift from object-oriented to procedural has to take place. Procedural is, after all, only an abstraction of the underlying sequential instruction processing of the von Neumann machine, which finally executes the application. Application development paradigms that are only object-oriented on the glass leave the responsibility for that paradigm bridging to the application programmers. More fully object-oriented approaches drive that bridge deeper into the system, with the operating system ultimately handling the shift—if, like OS/2, it is robust enough to handle it. Short of operating system implementations of the paradigm shift are the application execution environments that handle it, such as the Smalltalks where the shift is handled by the virtual machines.

Good ideas tend to percolate down through system layers, and the fact that object-oriented programming is already being implemented in the OS layers is another solid testament to its acceptance. The question of object-oriented microcode and/or hardware remains for future contemplation.

HOW THE SIZE OF PARTS SUGGESTS "COTTAGE INDUSTRY" VIABILITY

An interesting attribute of parts, especially in contrast with procedural code, is their tendency to remain small and specialized. In fact, some of the most interesting parts are the small ones because they embody *only* the author's domain skill.

It follows then that if some of the most useful and interesting parts are indeed fine-grained parts, this technology advance has the additional potential of reawakening and exploiting a vast reserve of domain-specific pro-

gramming expertise in a highly productive and reusable fashion. If useful parts can indeed be small, one person can conceive of, design, program, test, package, and profitably market useful functions. If the cost of entry into the business of turning one's domain expertise into useful parts is not too great, we can expect that many will do so on the individual and small-company level. Hence, the newest cottage industry could well be parts.

While many interesting parts are

small, that does not mean that large interesting parts don't exist. A new vocabulary is emerging, but without becoming a glossary, the sidebar on page 36 puts a few definition stakes in the ground, at least for purposes of this article.

While object orientation does not guarantee good coding, one axiom seems to be emerging: objects with long methods (their functional code) are not examples of good coding. Here's the crux of how objects will re-

Invest a stamp



Save a bundle

For the price of a stamp, you can get the latest edition of the federal government's free **Consumer Information Catalog** listing more than 200 free or low-cost government publications on topics such as federal benefits, jobs, health, housing, education, cars, and much more. Our booklets will help you save money, make money, and spend it a little more wisely.

So stamp out ignorance, and write today for the latest free **Catalog**. Send your name and address to:

Consumer Information Center
Department SB
Pueblo, Colorado 81009

A public service of this publication and the Consumer Information Center of the
U.S. General Services Administration.

GET DEBUGGED!

REXX Diagnostic Commander Makes REXX Debugging Easy

Software developers, administrators, and all OS/2 professionals can efficiently code and test REXX command files in a completely interactive environment. Easy to use, RDC enhances your code writing ability while increasing productivity. *Programs are debugged easily and quickly!*

- Supports Multiple Platforms
- Source Code Display
- Variable Display Service
- Variable Alteration
- Single Step Review
- Dynamic Watch Windows
- Logic Alteration
- Deferred Debug Option

**Get the REXX
Diagnostic Commander**

Only \$45⁰⁰

Order It Today!

**Call
800-724-7011**



**Suitable
Alternatives**

4655 Old Ironsides Drive, Suite 220
Santa Clara, CA 95054 • 408-727-3142



main small and therefore remain within reach of the cottage industry. In contrast with procedural code, object-oriented programs are enhanced by adding objects or methods, not by extending or increasing the complexity of existing ones.

HOW A SIMILAR PHENOMENA OCCURRED WITH THE ADVENT OF THE PC

In the early 1980s, on the heels of the popularization of the PC, some very popular, very small, shrink-wrapped programs appeared on the market quickly. These programs were the fruits of some enterprising small companies and individuals who could conceive, develop, test, package, and market interesting software. Machine sizes were not large. The original IBM "PC 1" had a 16K motherboard that quickly gave way to a huge 64K motherboard! The PC was new, and there was both novelty and utility in some very small programs. A cottage industry of software providers was growing.

This new cottage industry didn't last long, however, because an interesting program did not stay interesting if it remained small. The function, feature, and performance pressures of the software business soon forced the size of simple applications into the domain of application systems that had to be developed and managed by teams of people, not individuals. The cottages either grew into skyscrapers or they died away.

IBM's own Employee Developed Software Program was an example of this earlier software cottage industry. Many IBM employees, myself included, got PCs of their own, and, unlike me, many of them produced some interesting code. With this reservoir tapped for software production, the Employee Developed Software Program became an alternative for the hardest part of any small business: marketing and merchandising. However, for the same reasons that the early PC cottage software industry faded away, so did the IBM Employee Developed Software Program.

HOW THIS TIME THE CAUSAL FACTORS BEHIND THE DIE-OFF DO NOT APPLY

It appears to me that if you look at the reasons for the fading of the PC's earlier cottage industry side by side with the notion asserted earlier that interesting parts can include small ones, it

follows that the parts cottage industry will not die off like the earlier PC one did. The function/feature pressure precipitates more parts, not larger application systems. Further, object-oriented programming by its nature will constrain the size of parts, making their creation within the reach of the cottage. Reflecting this trend is the possible resurgence of the IBM Employee Developed Software Program.

HOW ALL THIS SUMS UP

The bottom line is that the software industry will undergo wrenching economic and cultural changes as well as technical changes in the coming months and years as a consequence of the shift to a parts economy. Facilitated and enabled by object-oriented technology, these changes will also apply to development technologies that are not strictly object-oriented. The market has already seen many development approaches in the 3GL and 4GL domains adopt parts strategies. Given these multidimensional changes in the wind, developers for platforms such as OS/2 will do best to anticipate and plan accordingly. As always, change means opportunity. How we deal with that opportunity is the challenge.

Richard K. (Rick) Runyan is a program manager for IBM's Software Solutions Division at Research Triangle Park, N.C. Rick manages the IBM Object Connection Program, a program for the encouragement of companies interested in going into the business of developing parts for IBM's object-oriented application development products and architectures such as IBM Smalltalk, VisualAge, VisualAge C++, System Object Model (SOM), Distributed System Object Model (DSOM), and OpenDoc. He can be reached via Internet at rick_runyan@vnet.ibm.com.

VisualAge, VisualAge C++, System Object Model, and OS/2 are trademarks of IBM Corp.

OpenDoc and Macintosh are trademarks of Apple Computer Inc.

Windows and Visual Basic are trademarks of Microsoft Corp.

Visual Works is a trademark of ParcPlace Systems Inc.

PARTS is a trademark of Digital Inc.

Taligent is a trademark of Taligent Inc.

Circle Reader Service Number 14

This article presents the process of developing a Lotus Notes application. It covers Notes basics, building blocks, and a general development methodology.

By STEVE KUPEC and CHARLES CARRINGTON

Beginning Application Development with Lotus Notes

Suppose you've been given the project of prototyping a Lotus Notes application; where do you start? Before we discuss how you might approach developing an application in Lotus Notes, let's describe Notes itself.

Lotus Notes is a client/server framework that implements the groupware concept via four main components: a document database, a communications platform, an application development environment, and a user interface.

The document database stores and manages compound object documents. It also provides the access methods for multiple user access and security. Notes does not depend on the operating system to implement

access control, security, or encryption.

The communications platform supports both mail routing and database replication. The replication process is unique to Notes; databases are automatically synchronized as documents are modified, added, and deleted. The integral communications supports LAN, WAN, and remote dial-up clients for every application.

The application development environment lets either a user or developer design applications that work with document databases. The development tools are included in every full client license.

There are three types of client licenses for Notes: Notes, Notes Desktop, and Notes Express. Notes Desktop and Notes Express



Steve Kupec



Charles Carrington



Figure 1. Notes Workspace—OS/2 client desktop.

do not include the Notes application development tools. Notes Desktop provides a run-time license for individuals who will be using applications developed by third parties or corporate developers. Notes Express provides a run-time license for individuals who will be using a small set of specific applications included with the Notes Express product.

The user interface is graphical, based on a folder concept; each folder can contain an assortment of applications and databases, as shown in Figure 1. Users point and click and drag and drop according to the rules of their particular workstation operating system.

Lotus Notes applications are built around these compound object document databases. A single application may reference more than one Notes database, but typically a Notes application refers to a single database. We will use these terms, database and application, interchangeably. To paraphrase Marshall McLuhan, "The database is the application."

Notes databases are used to store information in the form of documents. You will see rows and columns of data (a Notes "view") in the typical Notes database, but these represent a collection of documents rather than a collection of records, as in a DBMS. You will see compound object documents that can consist of rich text, images, graphics, audio, and video. The database also stores the basic building blocks of your application in the same physical space as the documents themselves. This is the first major shift you will need to make as you undertake application development under Notes (ap-

plications are ways of manipulating documents stored in a database).

WHAT IS NOTES GOOD FOR ?

Notes is a flexible, object store and forward, group-oriented data environment. It is appropriate for a wide variety of organizational tasks not normally given to databases. In order of increasing programming complexity, Notes is well suited for:

- Discussion databases
- Reference databases
- Workgroup or project tracking
- Forms-based applications
- Workflow applications.

Notes is not a relational database. Data is stored in what is, in essence, a flat file structure. The internal file structure does use an index to help organize the data. Notes doesn't attempt to duplicate what is available in the relational world. It does provide numerous links to external data sources. (Native function calls include @DbLookup and @DbColumn. References on other options are provided at the end of the article.) It does not support the concept of staged commits or rollback.

Notes is not a transaction processor. It does not support record locking (infeasible with distributed replica database copies). Because Notes data in different locations may be out of synchronization, it would not be appropriate for a scheduling or accounting application.

HOW TO BUILD A NOTES APPLICATION

Building blocks. What are the basic building blocks in Lotus Notes? The primary building block is a field. Fields can contain any type of data imaginable.

Fields are combined to build a form (Figure 2). Forms are not documents; they are simply one way of accessing the data stored in a document. Forms are used to manipulate an individual document. The next building block is a view, which is illustrated in Figure 3. Views are used to work with a collection of related documents. Let's look at an example.

Sample application. We will be using a "Customer Complaint" application as our example. The documents in this database are created when a customer calls our firm with a complaint. The operator pulls up a blank form on his or her screen. The form contains fields for customer name, address, phone, complaint subject, and so forth. We could even have a field that contains a bitmap of the fax the customer sent us, a copy of the trace, or a voice object recording of the actual customer complaint call. (Notes applications can provide 24-hour customer service!)

The fields are organized on the screen to reflect the appropriate data entry pattern. Notes provides the capability for data entry validation, default values, keywords, and table lookups in other databases (for example, for zip codes). The fields can be arranged and displayed differently for edit (data entry) modes and view (read only) modes in the same form. The operator can search the database for possible resolutions to a similar complaint using a full text search. He or she can also look at the existing data in a variety of ways. These are the views mentioned earlier (for example, Open Complaints by Product). If your application doesn't have the exact view the operator needs, he or she can create a private view to see the desired data.

Basic Tools. The standard Notes client user interface provides the tools you need to develop an application. Anyone familiar with a visual programming environment will be comfortable almost immediately in Notes. You can build forms, views, icons, and macros using the built-in functionality of Notes.

Templates. As a Notes application designer, you have several options for creating a Notes application. The easiest method would be to use one of the existing application templates delivered with the Notes client software. The client software comes with over two dozen templates and example databases you can use immediately to jump-start your development effort. Just copy the template and start entering your data.

Figure 2. What is a Form?

Another method of application delivery is to take a template and customize it by adding additional forms, views, and macros. Templates provide a generic starting point for developing an application. In fact, any existing application database can be used as a design starting point. It is easy to cut and paste parts of several existing applications to create a new application.

A third method would be to start with a blank canvas and build your application from scratch using the native functions and tools inherent in Notes itself.

Another option is to buy a third-party application for Notes; this is especially helpful if you need to use some of the advanced features of Notes (such as controlling work flow) but don't yet have the experience to do the coding yourself. (Examples include news services delivering reports customized to your company requirements, executive information systems providing graph charting and data drill down functions on mainframe data, or telephone-accessible databases.)

A fifth method of delivery would be to code an application that uses Notes' API. You could use C, Visual Basic, REXX, Lotus Notes HiTest tools for Visual Basic, or Lotus's own VisualProgramming (ViP) tool to do this. All of these development tools provide some level of access to the core Notes API. They allow an external process or non-Notes client to interact directly with a Notes database.

Choose the method that best suits your organization, but we suggest using a template as a starting point. Following are some advantages of using a template:

- You don't start a project staring at a blank screen.
- It allows you to reuse tested objects.
- It helps you standardize your organization's document style.
- It assists users who expect the applications to operate in the same basic ways. (A user will likely have tens or even hundreds of Notes applications on the Notes desktop, and the similarity reduces training and support costs.)

Managing Code. The next major paradigm shift you'll make deals with your application itself. Your code is stored in the database along with the documents. You can separate the storage of this code by creating a template of your application. A template is simply a copy of all the database structures and definitions, without any data.

One advantage to this structure is that every time the user (client) accesses the server for data, you have the opportunity to update the application code. You do not have to physically visit the workstation for updates. Such updates can be forced or voluntary. In fact, the users may not even know application updates have occurred.

In our application development efforts, we have at least one template for each application. A typical scenario has a Notes application stored on a server as a "*.NSF" (Notes Structured Format) file and a template as a "*.NTF" (Notes Template Format) file. Either file may also be stored on a local hard disk so development activities can occur off-line from a production server. Refer to Figure 1 to see how this would look on the Notes Desktop.

The NTF file does not have any real data associated with it; only test data is used with this file. The NSF file has the production data and the production version of your code. Set up an inheritance relationship between the template and the database where the database gets all of its structure from the template. To make a change to the application, start by updating the template (or a copy of the template). After testing your change with sample data, tell the database to inherit its (new) structure from the template, either immediately or at the next replication (scheduled or impromptu). Replication is an exchange of updates between discrete copies of a database.

OTHER PROGRAMMING DIFFERENCES

You will need to be aware of several other paradigm shifts as you approach

Notes application development. Notes does not have any provision for a source code control system or a data dictionary; you have to manage your templates and databases manually. Notes does provide a tool (Design Synopsis) that lets you capture the entire design of the database. We use a separate database to track changes to our production applications and store copies of the design synopsis.

There is no compilation process in Notes either. Your code exists as a collection of forms, views, and macros. This is both the source code and the executable. Notes does have a security hierarchy that ensures only certain people can make modifications to production versions of your application. The database manager is the final arbiter. This person controls who can develop or modify the application as well as who can use it for data entry or retrieval. A database designer can modify the forms, views, and macros but cannot change any of the access security features.

At the start of the article, we mentioned that Notes is a client/server framework. We say this because Notes is a cross-platform, multinetwork client/server environment. The Notes server code runs on IBM's OS/2 and AIX, on Novell's UNIX and as a Netware Loadable Module (NLM), on Sun's Solaris, on Hewlett-Packard's HP/UX, SCO UNIX, and on Microsoft's NT and Windows 3.x. The Notes clients run on IBM's AIX and OS/2, Microsoft's Windows, Sun's Solaris, SCO UNIX, and Apple's System 7.x. Other platforms are being added by Lotus on a regular basis.

Date Received	Assigned To	Request Type	Requester	Dept.
04-30-95	Carrington, Charles	New application development.	A. Customer	Finance
		System requirements (Steve Kupec, 04-30)		
03-01-95	Team, Change	Fix defect #124	Manager, My	Service
02-24-95	Dept. MIS	Look into new DASD for server.	Smith, Joe	Marketing
01-15-95	Carrington, C.	Can't Install	Corp. ABC	Help Desk

Figure 3. What is a View?



Several networking protocols are available to make the connection between the client and server. The most common are NetBIOS, TCP/IP, AppleTalk, Banyan Vines, and IPX/SPX. Notes also supports disconnected users via remote dial-in capability using standard modems and a default proprietary protocol, X.PC.

Notes application development can be done on any supported client platform, and the resulting database can be used immediately on any other client or server platform without any rewriting or recompilation.

BENEFITS AND SHORTCOMINGS

As you can see, Notes can provide you with a client/server system that spans your enterprise. You don't need to worry about coding to a specific platform; Notes supports a heterogeneous mix of workstation operating systems. Rolling out a client/server application with this type of framework becomes quite realistic. It addresses the often vexing problem of providing and maintaining a high-function application over a disparate set of users (and mix of operating environments).

Notes has other features that will make your delivery of applications easier. First, Notes has an elegant replication methodology. Your applications do not need to concern themselves with database synchronization between servers; Notes does that automatically, both for data and code. You can install a wide area network linking your servers as well. Your remote users can dial in and replicate the databases they need while on the road. Notes supports its own internal mail system as well as products like cc:Mail.

Notes does have its shortcomings. It has macros but no real script language. This may change with the next release of Notes (Version 4). Lack of any recursive or looping ability is a serious omission.

Notes mail is often disparaged as having a less elegant user interface than dedicated mail programs. In fact, Notes Mail is simply another database template that can be customized to suit the specific needs of the user and the organization. Mail communication functions are an integral part of Notes, and by default, all the Notes applications you create will be Mail-enabled. Expect further changes with Notes Version 4.

DEVELOPMENT OPTIONS AND EXTENSIONS

In addition to the capabilities inherent in Notes itself, there are several other options for delivering Notes functionality. Lotus and third parties have complementary products that significantly enhance the range of applications you can develop. Lotus' Notes API Toolkit is a C language subroutine library that provides you with access to Lotus Notes services. The Toolkit contains header/library files, documentation, and example programs.

Lotus' VisualProgramming (ViP) is a visual development environment (Windows only), similar to Visual Basic, that allows you to create custom graphical user interfaces to the Notes environment and provides easy access to both legacy and relational database systems. Lotus' Forms (Windows only) is used for designing, testing, and implementing form workflow applications in workgroups. It supplements the workflow capability in Notes itself.

Lotus' Phone Notes is a development environment specifically tailored for the creation of telephony applications under Notes. Lotus' Notes HiTest is a collection of Windows development tools designed to help Visual Basic programmers work with Notes. Lotus Notes: Document Imaging (LN:DI) is a document imaging extension for Notes. IBM markets a REXX API toolkit called InterfloX, which gives you a REXX interface to Notes services. For more information about InterfloX, please see Tony Walsh's article on page 53.

JUST DO IT

The best way to get started with Notes is to just do it. Get yourself a client copy and build an application. Look at the sample databases for hints. Use the included Application Developer's manual when you get stuck. Look at the tutorial. Start stretching the Notes limits. Then begin exploring methods to link Notes to your existing data sources—your user's won't be far behind.

Steve Kupec, who has a B.S. from S.U.N.Y. at Albany and is a M.S. Candidate at Marist College, has been involved with Notes for nearly two years. His background is application development, project leadership, and client/server support. He was involved with Notes application development and the Notes rollout at IBM's Personal Software Products

division in Austin, Texas. He is the president of a Lotus Notes users group in Austin and can be reached at kupec@austin.ibm.com or Steve Kupec @ AUSNOTES @ LONESTAR @ NOTES NET.

Charles Carrington has a B.A. and an M.B.A. from the University of Texas and has been active in Notes applications development since 1993. He operates an independent consulting business, concentrating in Lotus Notes activities, in Austin, Texas. Clients include IBM PSP. He is an active member of the Lotus Notes user group in Austin. He can be reached at ccarrington@tpoint.net.

REFERENCES

Many resources are available to help you get started with application development in Notes. We suggest the following:

- Lotus's home page at <http://www.lotus.com>. Here you will find press releases, communiqués, and technical reports.
- Lisa Pyle's book, *Creating Lotus Notes Application*, QUE Publishers, ISBN 1-56529-556-0, \$29.95. This is a good introductory text on application design.
- Tony Walsh's book, *Notes in the OS/2 Environment*, John Wiley & Sons, ISBN 0-471-13127-X. This is the best book we've seen on Notes and OS/2 (see article on p. 53).
- The following newsgroups on the Internet:
 - comp.groupware.lotus-notes.misc
 - comp.groupware
- A Lotus Notes users group in your city. We are active in a users group in Austin and find this a valuable way of getting information about how to effectively use Notes.
- Take a class from a Lotus Business Partner on application development.

Software Development '95 East

October 2-6, 1995 Washington, D.C.



Your Mind is Your Livelihood. Feed it at SD '95.

Software Development '95 is the most important event of the year for cutting-edge development professionals.

Interact with 10,000 of your peers and condense months of research and learning into a few short days. Over 150 lectures, workshops and tutorials featuring the industry's leading experts cover today's most relevant development topics:

- Team Managers and Management
- Managing for Quality and Productivity
- C++
- Windows 95/NT Development
- UI Design
- Building a Software Component Strategy
- Testing and Debugging
- Enterprise Application Re-Development
- Client/Server Architectures and Tools
- Database Programming and Design
- Object-Oriented Programming
- Building Open & Distributed Systems



Also see over 300 development tools from over 200 leading vendors and have your tough questions answered by the product experts.

Contact us today for the opportunity to increase your productivity, enhance the quality of your software, and feed your mind at SD '95.

Circle Reader Service Number 41

Fax:
(415) 905-2222

Call:
(800) 441-8826

E-Mail:
sd95east@mfi.com

☐ Yes! Send me more information on Software Development '95 East.

Name

Title Company

Address

City State Zip

Phone Fax E-mail

Home Page <http://www.mfi.com/sdconfs/>

Reply today!

Enter your name
for a chance to
win a FREE
SD '95 VIP
Conference Pass.

OS2D



Object-Oriented Programming

This article presents some design tips and strategies for designing applications using distributed object technology. It is intended to help developers increase the performance and flexibility of their distributed object applications. By F.R. CAMPAGNONI, PH.D.

Distributed Objects: Application Design Tips

Distributed object technology is a hot topic right now for folks interested in building network applications—and who isn't? Some of these applications will be successfully deployed and some will not. At least a portion of the failures will result from ineffective application design. Believe it or not, it is possible to build applications that are brittle and/or perform poorly even when using object technology. And with distributed object technology, the considerations that must go into building applications that span address space boundaries make the design problem a bit trickier.

This article is intended to give the reader some tips on the effective design of applications built using distributed object technology. It is not intended to be an academic treatise on the science of leveraging distributed technology but rather a presentation of some design strategies and heuristics that can help you avoid common pitfalls. I offer them as a resource to developers struggling to build efficient, flexible applications using distributed object technology, particularly to those working with my favorite distributed object technology: IBM's SOMobjects Toolkit.

WHAT IS DISTRIBUTED OBJECT TECHNOLOGY AND WHY SHOULD I CARE?

Distributed object technology is an enabling infrastructure and set of system services that allow a developer to build an application from interconnected objects located in different address spaces on the same machine and/or different networked machines.

A central element in the infrastructure is something called an Object Request Broker (ORB). The ORB is the underlying mechanism by which objects can make requests on one another in a uniform fashion without regard to the language in which the target ob-

ject is implemented, the operating system on which the object is running, or the location of the object on the network.

A consortium called the Object Management Group (OMG) is working to standardize the programming interfaces across and to provide interoperability between different ORB implementations. IBM's ORB implementation, which is part of the SOMobjects Toolkit for OS/2, AIX, and Windows, adheres to the OMG standards frequently referred to as CORBA (Common Object Request Broker Architecture) or COSS (Common Object Services Specification). Currently, fully interoperable versions of SOM are available for OS/2, AIX, and Windows 3.1. Soon, Macintosh System 7, OS/400, and MVS will be added to the list.

Distributed object systems are intended to make leveraging network services as simple and transparent as possible for the developer. ORBs provide an object-based abstraction above the communications layer that hides that layer's complexity. The form and semantics of a method invocation on an object are identical regardless of whether the object on which the request is being made is in the same location as or a different location from the requester.

While distributed object systems make it much easier to build distributed applications, this does not mean you can ignore distribution when designing applications. If enough attention is not paid to the organization and relationships between and within the distributed pieces, it is actually pretty easy to build distributed object applications that perform poorly and are difficult to change.

SOME GENERAL DESIGN CONSIDERATIONS

Two common focuses for designing distributed applications are performance and partitionability.



Performance tends to be important for any application, but it is particularly an issue with distributed applications because of the tremendous overhead involved in communicating outside of a single address space. As a result, calls across address spaces need to be carefully considered with regard to their performance consequences. Some of the performance issues are discussed ahead.

Partitioning is the act of breaking an application into pieces that execute in different address spaces either on the same machine or across machines. Applications are partitioned for many reasons: to increase robustness and fault tolerance, enforce security, and off-load compute-intensive operations onto a server or leverage services available on other machines or platforms. In general, the more partitionable an application is, the more flexible it is to deploy. This flexibility means the application can adapt quickly to changing conditions such as the availability of new hardware or network resources.

With distributed object technology, this process is made somewhat easier because the application is constructed from self-contained components or objects. It is a more natural process to partition objects or sets of objects into address spaces. Partitioning can't be done arbitrarily, however, hence the first tip:

Start with a design. This probably sounds trite, but it is critical to ensuring that the implementation of the application will have the desired performance and flexibility. Before beginning to write the final code for your application, use an object-oriented design tool or other diagrammatic aid to document the application design. It is particularly important to clearly document the calling and argument-passing relationships between objects and object subsystems. This information will provide the basis for an analysis of application performance and partitionability.

This may seem somewhat at odds with the notions of prototyping and interactive development that are synonymous with object-oriented development. However, iterative development and prototyping can (and often should) be orthogonal processes.

The prototype is the experiment—the proof of concept. It is where one tests the assertion that the application is practical to build at all. During prototyping, it is easy to meander down the garden path. One makes ad hoc decisions and cuts corners in order to proceed quickly. This can lead to overly complex and convoluted relationships be-

tween elements in the application. It is important at some point to back away from the prototype and take a critical look at the design that has emerged and consider whether it can be simplified and optimized.

Iterative development begins with a working kernel of an application and adds functionality in a planned fashion while continually maintaining a working version of the application. This is the spiral, rather than waterfall, model of software development. During iterative development, it will no doubt be necessary to fine-tune the design, but one is unlikely to encounter an issue that will require a radical redesign of the application if it has been properly prototyped first.

Think frameworks, not objects. A framework is a set of interrelated objects that, working together, perform a function (for example, three-dimensional graphics or device I/O). In general, it is useful to visualize your application as a set of cooperating pieces or subsystems. Each subsystem provides an external abstraction for use of that subsystem by other parts of the application. The design of this abstract interface is important because it determines how easy it will be to replace the existing subsystem with one that provides the same functionality but is implemented differently. An application constructed from well-designed frameworks can be adapted quickly because it is relatively easy to pop out an existing implementation and substitute a different one (with the same programming interface).

Frameworks also can provide natural boundaries for partitioning. Objects within these subsystems tend to be tightly interrelated or coupled with each other and loosely coupled with objects in external components. More on this later.

Understand and attend to boundary conditions. This is an important principal from the software testing domain that applies to application design. Essentially, it says "expect to find trouble where you push the limits of your software system." If your application depends upon robust execution on boundary conditions, be prepared to lavish additional time on design, implementation, and testing and the inevitable problems that will arise.

With distributed object-based applications, one runs into boundary conditions doing such things as passing very large amounts of data or extremely complex data types in a method call, crossing "seams" be-

tween object-oriented and non-object-oriented systems, or using interoperability protocols or gateways to cross between different distributed object systems.

Boundary conditions are where problems are most likely to arise. As a general rule, it is better to work well within the capabilities of the system by doing things in the simplest, most straightforward fashion (KISS—keep it simple stupid). That doesn't mean you shouldn't push a software system to its limits when necessary. However, it does mean that you must be cognizant that in doing so you are more likely to encounter problems. And it is important to keep your expectations reasonable. For example, unless the appropriate level of benchmarking is done, it is unrealistic to expect as high a level of performance from a call to a remote object that passes a large amount of data as arguments as a call that passes a few simple integers.

DESIGNING FOR PARTITIONABILITY

Following are some guidelines for building applications from components that can be distributed (and redistributed) across address space and machine boundaries.

Encapsulate platform-specific functionality. Calls to platform-specific functionality (for example, window system calls) should be isolated into a set of objects. There are two advantages to this approach. First, objects that do not make platform-specific calls can theoretically run on any platform. These platform-independent objects can be flexibly partitioned (and repartitioned if required) across different machines.

Second, isolating platform-specific calls into a set of objects that can be replaced as a unit makes it easier to port an application to a platform other than the one on which it was originally written.

Separate the presentation components of an application from the semantic components.

The presentation/semantic separation of components is sometimes called the model/view or document/view separation. These terms refer to the fact that it is convenient to divide application functionality between components concerned with user interface and rendering issues (application presentation) and those concerned with things like data access and calculations (application semantics).

One reason this is important for building distributed applications is that, unless the developer is using a distributed window system like X-Windows, presentation components will need to reside on the same machine as the display, keyboard, and mouse. However, properly designed semantic components can be deployed across different machines. The careful separation of application functionality into presentation and semantic components will yield higher levels of partitionability and adaptability.

Argument passing considerations for partitionability. If an interface is expected to be called from another address space, it must be designed with remote access in mind. For example, the CORBA specification does not support passing pointers across address spaces. As a result, an interface specification that contains a pointer is not CORBA-compliant and will not work for many CORBA implementations.

In the next release of SOM, pointers to some datatypes can be passed across address spaces. In these situations, the SOM run time does the work of copying the data the pointer is addressing and passing it to the destination address space so it is available during method execution. However, that can still be a problem if the pointer addresses a huge data structure or large block of data. This data would need to be copied between the address spaces, and doing so could have undesirable performance implications (discussed ahead).

Carefully design interfaces that can potentially be called remotely. Make the arguments and return values to these methods as simple as possible and avoid datatypes that cannot be efficiently passed across address spaces.

Analyze the "uses" relations between components. The uses relations between components refers to the calling patterns of method invocations between objects. If the application design has been properly documented, as previously suggested, it can be used to analyze the calling relations between objects (the call graph). The call graph can be very helpful in identifying areas where the design can be simplified. The strategy here is to avoid complex and convoluted calling relationships between components in an application. These loops in the uses relations create interdependencies between components that make an application difficult to partition and difficult to change.

Design around affinity groups. An affinity group is a set of components

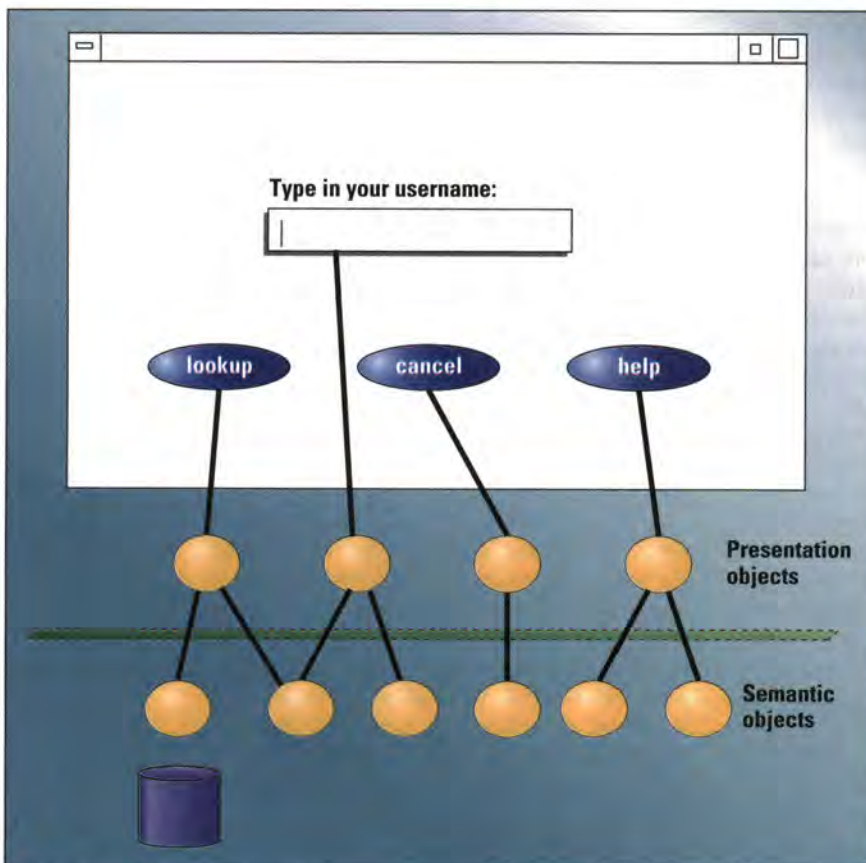


Figure 1. Separating presentation from semantic functionality makes an application easier to partition and change. Properly designed semantic objects can be driven by many different types of presentation objects.



that are closely interrelated. This usually means that they make frequent calls on one another and less frequent calls on components in other groups. Affinity groups are similar to frameworks in that they have strong internal and weak external coupling. The difference between affinity groups and frameworks is that affinity groups tend to be higher-level subsystems that can contain multiple framework and nonframework components.

The strategy is to design your application around sets of closely interrelated components that are strongly coupled with each other and weakly coupled with other components or affinity groups (Figure 2). Designing around affinity groups enhances the partitionability and adaptability of an application because, like frameworks, affinity groups can be replaced as a unit and tend to provide natural boundaries for partitioning.

DESIGNING FOR PERFORMANCE

Although the form of the method call a programmer writes is the same regardless of whether the target object is in the same address space, in a different address space on the same machine, or on a different machine, the code path the execution of the call makes is quite different. Which code path is executed is invisible to the developer, but it makes a big difference in terms of how long it takes to execute a method on an object.

Using SOMobjects, a call to an object in the same address space is handled much the same as a local function or procedure call. For calls to an object in another address space on the same machine, SOM uses local interprocess communication (IPC) mechanisms for notification and argument passing. Local IPC mechanisms are almost always faster than communicating between two processes on the same machine through the network adapter. Finally, for calls between objects on different machines, SOM relies on whatever network transport mechanism it has been configured to use.

Following are some guidelines for optimizing the performance of distributed object applications.

Design around performance rings.

Because the form of the calls are the same regardless of where the target object is located, it is easy to forget that the cost of executing a method on an object in another address space from the caller can be many thousands of times greater than executing a method on an object in the same address space.

When you cross an address space boundary, you move from microsecond to millisecond method execution speeds.

This is not a characteristic of distributed object systems, per se, but of all distribution systems. It is caused by the communications overhead necessary to coordinate code execution between components of an application that do not share the same execution context. For example, traditional remote procedure (RPC) calls are many thousands of times more expensive than local procedure calls.

For the purpose of comparison, here are some rough ballpark figures of using SOM:

- Cost of a within-process method call is approximately equivalent to a procedure or function call.
- A method call that crosses an address space boundary into another address space on the same machine is 1,000 to 10,000 times more expensive than a method invocation within a single address space (function call).
- A method call that crosses an address space boundary into another address space on a different machine is three to five (or more) times

more expensive than a method call across address spaces on the same machine. This of course depends on such factors as network load and the number of "hops" (routers) between source and destination nodes.

This creates what I call performance rings (Figure 3) and suggests a strategy for designing distributed object applications that are optimized for performance. In general, the more frequently two objects need to communicate, the closer to the center of the concentric performance rings they should be located with respect to one another. This means that objects that need to make frequent calls on each other should be in the same address space. Of course, there are always exceptions to this rule, but they need to be carefully weighed with respect to their performance consequences.

Argument passing considerations for performance. Two important dimensions should be considered in arguments passed to method calls on remote objects: the amount of data being passed and the complexity of the data being passed. Of the two, the most important consideration for performance is the amount of data being passed. This is because passing data between

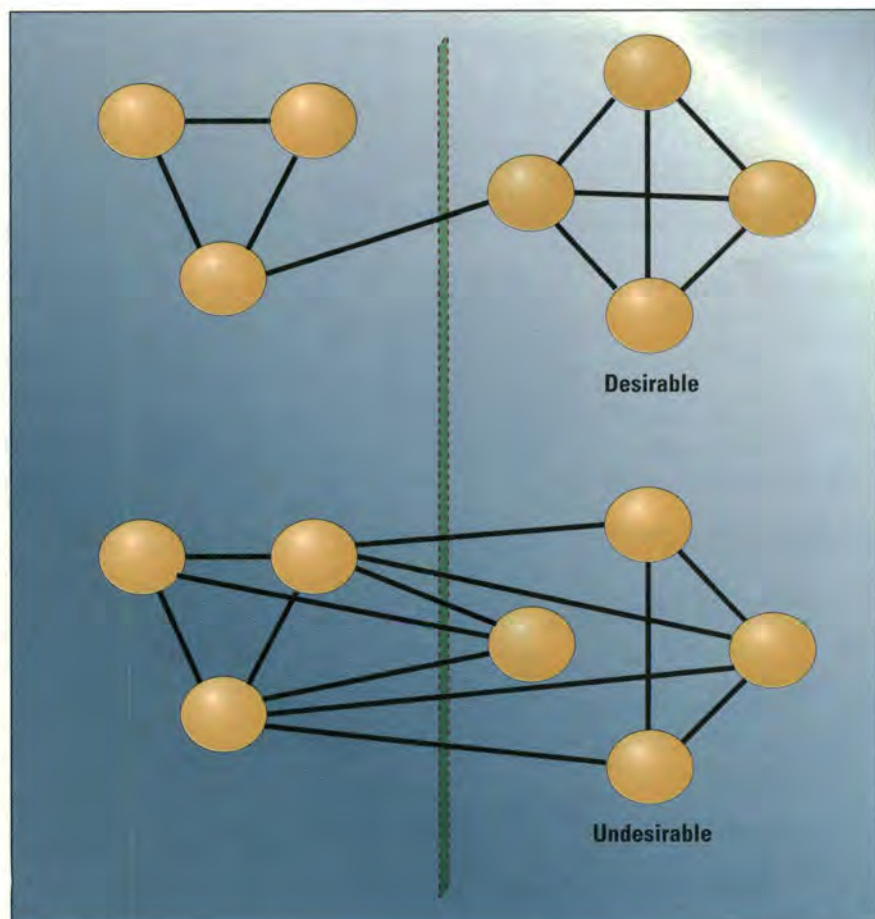


Figure 2. Design around affinity groups with tight internal and loose external coupling.

What does it take to test a client/server system?

Synchronized, multi-user testing
that goes beyond the types of testing possible with stand-alone, GUI test tools.

Unattended, extended use testing
that lets you determine how your app stands up to continued pressure.

Stress, load, and performance testing
to answer key questions about the activity level your system supports.

The Softbridge Automated Test Facility has been testing client/server systems since 1990. To learn how ATF can help you test OS/2, Windows, and NT apps, call 617-576-2257 (or fax 617-864-7747).

ATF The Softbridge
Automated Test Facility

Softbridge, Inc. ▲ 125 CambridgePark Drive ▲ Cambridge MA 02140

Circle Reader Service Number 15

IBM LAN Server? You Must Have "LAN Intensive Care Utilities"

Without it, what if...

- Alias logon assignments fail?
- Domain/accounts corrupted?
- Permission disappear?

How do you...

- Break-up/consolidate domains?
- Move accounts/resources?
- Administer hundreds of accounts?

See us at OS/2
World in
Boston, MA
Booth 706

You're dead without our tools!

Lieberman and Associates
221 N. Robertson Blvd. Suite C
Beverly Hills, CA 90211-1703
Phone: 310-550-8575
Fax: 310-550-1152 BBS: 310-550-5980
CS: 76426,363 IBMMAIL: USMVHLVH

Call: 800-829-6263

CALL OR
FAX FOR
IMMEDIATE
INFO!

Try it out yourself today!
Download it from: Our BBS,
Compuserve: GO OS2DF2 Lib3,
IBMLINK, Hobbes



Circle Reader Service Number 16

ADVERTISER INDEX

Reader Service	Company Name	Page
23	ChipChat.....	50
17	Compuware.....	64
31	Crossley Group	51
33	DB Technologies	52
9	Development Technologies.....	23
21	Essex Systems.....	50
19	Hockware.....	1
11	Indelible Blue	29
—	Inmark	COV4
24	Irmware.....	50
34	Jagre.....	52
27	Keller Group.....	51
8	ColoradOS/2.....	20
16	Lieberman.....	48
20	LPC Consulting.....	50
1	Metaware	COV2
5	MHR	9
10	MicroEdge	25
	Microwork	50
7	Mitnor.....	19
32	NetLinks Technology	52
12	Nimbus.....	35
3	Object People.....	4
29	Objectshare	51
25	Opt-Tech Data Processing.....	50
28	Perez Computing.....	51
2	Programmer's Paradise	2
4	Rimstar	7
30	Soffront.....	51
15	SoftBridge	48
22	Softel.....	50
14	Suitable Alternatives	38
6	Watcom	11
18	Zinc	COV3

Return these cards to

receive **FREE** information on

the products advertised in

OS/2 Developer. Simply

circle the reader service

numbers of the items you

are interested in, complete

the address information,

and mail the card.

To enter a paid subscription

to OS/2 Developer, check

the "YES" box, sign and

date the card, indicate a

method of payment, and

drop the card in any mailbox.

Return postage is paid.

Subscribe to OS/2 Developer

☐ **Yes** I want to start my subscription to OS/2 Developer for only \$39.95!

Signature _____ Date _____

☐ Charge my credit card: ☐ Visa ☐ Mastercard ☐ American Express

Card No. _____ Exp. Date _____

☐ Bill me.

* Canada and Mexico add \$16.00; Outside North America add \$30.00 for shipping.

Name _____

Title _____

Company _____

Address _____

City _____ State _____ Zip _____

Phone Number _____ Fax Number _____

Subscribe to OS/2 Developer

☐ **Yes** I want to start my subscription to OS/2 Developer for only \$39.95!

Signature _____ Date _____

☐ Charge my credit card: ☐ Visa ☐ Mastercard ☐ American Express

Card No. _____ Exp. Date _____

☐ Bill me.

* Canada and Mexico add \$16.00; Outside North America add \$30.00 for shipping.

Name _____

Title _____

Company _____

Address _____

City _____ State _____ Zip _____

Phone Number _____ Fax Number _____

Subscribe to OS/2 Developer

☐ **Yes** I want to start my subscription to OS/2 Developer for only \$39.95!

Signature _____ Date _____

☐ Charge my credit card: ☐ Visa ☐ Mastercard ☐ American Express

Card No. _____ Exp. Date _____

☐ Bill me.

* Canada and Mexico add \$16.00; Outside North America add \$30.00 for shipping.

Name _____

Title _____

Company _____

Address _____

City _____ State _____ Zip _____

Phone Number _____ Fax Number _____

Free Information

Circle the reader service number of those items of interest and drop this card in the mail. Your information will be sent directly to you. See Index for Reader Service listing.

1	11	21	31	41	51	61	71	81	91	101	111	121	131	141	151	161	171	181	191
2	12	22	32	42	52	62	72	82	92	102	112	122	132	142	152	162	172	182	192
3	13	23	33	43	53	63	73	83	93	103	113	123	133	143	153	163	173	183	193
4	14	24	34	44	54	64	74	84	94	104	114	124	134	144	154	164	174	184	194
5	15	25	35	45	55	65	75	85	95	105	115	125	135	145	155	165	175	185	195
6	16	26	36	46	56	66	76	86	96	106	116	126	136	146	156	166	176	186	196
7	17	27	37	47	57	67	77	87	97	107	117	127	137	147	157	167	177	187	197
8	18	28	38	48	58	68	78	88	98	108	118	128	138	148	158	168	178	188	198
9	19	29	39	49	59	69	79	89	99	109	119	129	139	149	159	169	179	189	199
10	20	30	40	50	60	70	80	90	100	110	120	130	140	150	160	170	180	190	200

July / August 1995; Expires January 1996

RSCG95

Free Information

Circle the reader service number of those items of interest and drop this card in the mail. Your information will be sent directly to you. See Index for Reader Service listing.

1	11	21	31	41	51	61	71	81	91	101	111	121	131	141	151	161	171	181	191
2	12	22	32	42	52	62	72	82	92	102	112	122	132	142	152	162	172	182	192
3	13	23	33	43	53	63	73	83	93	103	113	123	133	143	153	163	173	183	193
4	14	24	34	44	54	64	74	84	94	104	114	124	134	144	154	164	174	184	194
5	15	25	35	45	55	65	75	85	95	105	115	125	135	145	155	165	175	185	195
6	16	26	36	46	56	66	76	86	96	106	116	126	136	146	156	166	176	186	196
7	17	27	37	47	57	67	77	87	97	107	117	127	137	147	157	167	177	187	197
8	18	28	38	48	58	68	78	88	98	108	118	128	138	148	158	168	178	188	198
9	19	29	39	49	59	69	79	89	99	109	119	129	139	149	159	169	179	189	199
10	20	30	40	50	60	70	80	90	100	110	120	130	140	150	160	170	180	190	200

July / August 1995; Expires January 1996

RSCG95

Free Information

Circle the reader service number of those items of interest and drop this card in the mail. Your information will be sent directly to you. See Index for Reader Service listing.

1	11	21	31	41	51	61	71	81	91	101	111	121	131	141	151	161	171	181	191
2	12	22	32	42	52	62	72	82	92	102	112	122	132	142	152	162	172	182	192
3	13	23	33	43	53	63	73	83	93	103	113	123	133	143	153	163	173	183	193
4	14	24	34	44	54	64	74	84	94	104	114	124	134	144	154	164	174	184	194
5	15	25	35	45	55	65	75	85	95	105	115	125	135	145	155	165	175	185	195
6	16	26	36	46	56	66	76	86	96	106	116	126	136	146	156	166	176	186	196
7	17	27	37	47	57	67	77	87	97	107	117	127	137	147	157	167	177	187	197
8	18	28	38	48	58	68	78	88	98	108	118	128	138	148	158	168	178	188	198
9	19	29	39	49	59	69	79	89	99	109	119	129	139	149	159	169	179	189	199
10	20	30	40	50	60	70	80	90	100	110	120	130	140	150	160	170	180	190	200

July / August 1995; Expires January 1996

RSCG95

Free Information



BUSINESS REPLY MAIL

FIRST CLASS MAIL

PERMIT NO. 612

PITTSFIELD, MA

POSTAGE WILL BE PAID BY ADDRESSEE

OS/2 Developer

P. O. BOX 5252

PITTSFIELD MA 01203-9317

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES



Free Information



BUSINESS REPLY MAIL

FIRST CLASS MAIL

PERMIT NO. 612

PITTSFIELD, MA

POSTAGE WILL BE PAID BY ADDRESSEE

OS/2 Developer

P. O. BOX 5252

PITTSFIELD MA 01203-9317

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES



Free Information



BUSINESS REPLY MAIL

FIRST CLASS MAIL

PERMIT NO. 612

PITTSFIELD, MA

POSTAGE WILL BE PAID BY ADDRESSEE

OS/2 Developer

P. O. BOX 5252

PITTSFIELD MA 01203-9317

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES



address spaces is an expensive operation. And if a method is being invoked on an object on another machine, passing large amounts of data can require multiple network packets, which will substantially reduce performance.

A secondary consideration in argument passing is the complexity of the data types of the arguments. Some of the CORBA-defined datatypes can be quite complex (for example, sequences of any of sequences of structures). Marshaling complex arguments to send across the network is more CPU-intensive than marshaling simple arguments such as integers.

The rule of thumb for argument passing for distributed objects is to minimize the amount of data passed and, whenever possible, to make the arguments as simple as possible.

SUMMARY.

Hopefully, this article has provided some food for thought in the design of distributed object applications. This is by no means a comprehensive guide to designing distributed object applications, but I hope I have shown that by attending to some relatively straightforward characteristics of an application's design, one can maximize the

benefits of using distributed object technology.

F.R. Campagnoni, Ph.D. is an object architect for the Personal Systems Div. of IBM in Austin, Texas. Frank has served as IBM's representative to the Object Management Group's Object Request Broker task force and was instrumental in focusing IBM's attention on distributed object technology. Frank was one of the technical leads for the team that developed SOM's CORBA-compliant distributed object framework (distributed SOM). The opinions Frank expresses in this article are his own and do not necessarily represent those of IBM Corp. His Internet address is: frc@austin.ibm.com.



Figure 3. Performance rings.

REFERENCES

Object Management Architecture Guide, Version 2.0, Object Management Group, Framingham, Mass. 1992.

The Common Object Request Broker: Architecture and Specification, Version 1.2, Object Management Group, Framingham, Mass. 1994.

Common Object Services Specification, Vol. 1, Object Management Group, Framingham, Mass. 1994.

The SOMobjects 2.1 User's Manual (online), Shipped with the SOMobjects 2.1 Toolkit.

F.R. Campagnoni, "IBM's System Object Model," *Dr. Dobbs's Special Report, The Interoperable Objects Revolution*, Winter 1994/1995.



OS/2 SMALLTALK DATABASE ACCESS

- ODBTalk accesses any ODBC-compliant data source
- full ODBC SDK 2.0 support, native ODBC data types
- online documentation, source included, no runtime fees
- programming examples and sample application
- OO to RDBMS mapping framework, based on types & brokers, ideal for complex client-server applications
- versions for Digtalk and IBM Smalltalk products

LPC Consulting Service phone: 416-787-5290
937 Briar Hill Avenue fax: 416-787-9214
Toronto, ON M6B 1M1 CIS: 73055,123

Circle Reader Service Number 20

TCP/2TM

TCP/IP for OS/2[®]

Our customers say our tcp/ip
is the best because
it's faster, more complete,
and reliable.

Find out for yourself!

Supported: all released versions of OS/2
including 32bit API for 2.x and above;
SLIP and/or PPP on COM1 and/or COM2;
Ethernet; Token Ring; ARCNet; NDIS and ODI

(800) OS2-TCP2

Essex Systems, Inc.

One Central Street, Middleton, MA 01949

A Dan Lanciani Product

TCP/2 is a Trademark of DLD consulting. All other products and trademarks are the property of their owners. TCP/2 is Based in part on work done by the University of California Berkeley.

Circle Reader Service Number 21

Network Remote Control And Monitoring Help Desks Automated Agent Monitoring

PM/Pirate

Named Pipes
TCP/IP
Lowest network load
Record and Playback
All PM Video Modes
No Special Drivers
FAST!

Automated Cycling

\$89.00 + s&h



Digital Communications Solutions
2050 North Towne Ct. NE Suite 7
Cedar Rapids, IA 52402

Softtel also provides:

In and Outbound call center applications -
Voice information systems
Predictive dialing
Voice Mail
Internet Service provider and Client Software.
Presentation Manager Development and Consulting

(800) 701-7205

Circle Reader Service Number 22

Send text messages to pagers directly from OS/2!



- Easy-to-use workplace shell application
- Includes a command line interface
- You can 'page-enable' your applications!
- 32-bit, SOM-based code - Only \$79

ChipChat[®] Wireless Communicator

ChipChat-Cawthon Software

Dearborn, Michigan USA & Fukuoka Japan

Phone 313-565-4000 Fax 313-565-4001

Circle Reader Service Number 23



EDPT by IRMWare

Save Time & Money - Schedule Training, Assess Skills, Match Skills to Jobs, Plan Career Paths, Recommend Training Plans, Match People to Jobs. Use only the features you need!

EDPT - Employee Development Planning and Tracking a low cost solution to automate Human Resource procedures. PC or LAN from \$199.

Interfaces to Personnel, General Ledger and Budgeting Systems.

OS/2 and Windows versions NOW AVAILABLE

FREE PRODUCT DEMOS, BROCHURES OR CONSULTING INFORMATION

(602) 730-1510 or Fax (602) 413-0399/P.O. Box 51093 Phoenix, AZ 85076-1093

Circle Reader Service Number 24

Opt-Tech Sort/Merge

High Performance Sort/Merge/Select Utility

Use as a stand-alone utility or call as a Subroutine. Many features including unlimited file size, multiple keys, record selection, duplicate elimination, summing and more!

Call for free brochure

Opt-TechData Processing
P.O. Box 678

Available for:

Zephyr Cove, NV 89448

OS/2 or Unix \$249

Phone (702) 588-3737

DOS or Windows \$149

Fax (702) 588-7576

Circle Reader Service Number 25

JOB SCHEDULING SERVER

OS/2 PM 32-bit advanced job scheduler for NOVELL and IBM LANs. Schedules multiple OS/2, DOS, and Windows jobs for concurrent execution on specified or auto selected LAN nodes. Conditional job scheduling (depends on files and other jobs return codes), security, sophisticated calendar, job logging, priority, user APIs, job recovery, periodic scheduling, and much more.

Microwork, Inc.

Phone: (708) 940-8979

Fax: (708) 940-8979

FAX Developer Tools

From the developers of **FaxWorks for OS/2**
Client/Server API and Printer Driver Toolkits
Support for LANs, up to 96 lines per CPU,
and all popular fax hardware

Keller Group Inc.

Voice: (612) 429-7273
Email: kgroup@ibm.net

Fax: (800) 329-3293
Fax: (612) 653-1987

Faxworks is a trademark of Global Village Communication.

Circle Reader Service Number 27

TRACK Bugs, Projects, Releases, Customers,...



SOFFRONT TRACK for Windows™ is a customizable multi-user database to track bugs, customers, projects, versions and more.

It has a powerful query and reporting, change history, automatic notification, security, import/export, version manager integration, and total customization to meet your needs.

SOFFRONT Software, Inc. **1-800-SOF-FRONT** / 408.263.2703 (8-5 PST)

238 S. Hillview Drive, Milpitas, CA 95035 Fax: 408.263.7452

Circle Reader Service Number 30

The easy to use 32 bit OS/2 PM IPF Language Editor...

IPF Editor

- Add help to applications in minutes
- Designed for easy use by programmers and non-programmers
- Generate C and Resource source files to add help to applications automatically
- Import wordprocessing files, including WordPerfect, quickly and accurately
- Preview IPF output without compiling
- Create all hypertext links automatically
- Optional Voice Support Module
- Includes complete on-line help and documentation

IPF Editor: \$150
Voice Support Module: \$50

Orders:
1-800-IPF-7622

Information: (360) 428-5025
on compuserve at
70410,2416

Perez Computing Services
4725 Monte Vista Place
Mt Vernon, WA 98273

Circle Reader Service Number 28

VisualAge & PARTS

Spreadsheets, business graphics, tables, and other powerful controls are easy to add to your VisualAge™ and PARTS™ applications. **WidgetKit™/Professional** provides spreadsheets, bitmap viewers, virtual spreadsheets, printing, and more.

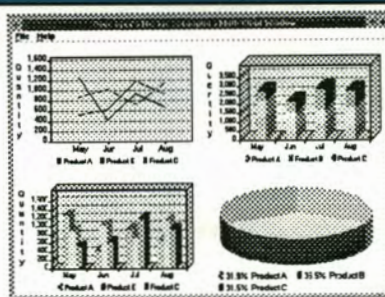
WidgetKit/Business Graphics has versatile graphs and charts.

Subpanes/V for PARTS has a powerful table pane, columnar list box, heirarchical list box, 3-D frames, 3-D and bitmap buttons, and more.

No royalties for end-user applications, 90 days free support, and 30-day money-back guarantee. Call or fax for free info.

Objectshare Systems, Inc. v: 408.970.7280 f: 408.970.7282
5 Town & Country Village, Suite 773, San Jose, CA 95128

Circle Reader Service Number 29



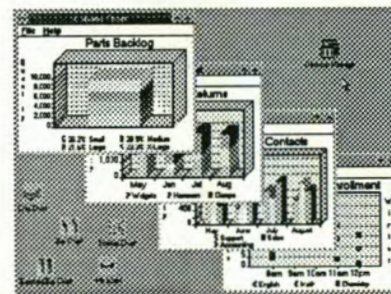
It's easy
to integrate
business graphics
into your
OS/2 PM®
or
Windows 3.x®
application

Developer's Business Graphics Toolkit

- High-level APIs
- Tactile feedback
- Real-time charting
- 16-bit and 32-bit
- Dynamic Link Library
- Positive/negative data
- Printer/metafile support
- 25+ chart types



The Crossley Group
P.O. Box 921759
Norcross, GA 30092
USA (404) 751-3703
Int'l 44-932-844-281
(demo available)



Circle Reader Service Number 31



Coming in September/October:

Database Development



Bonus Distribution:

Software Development '95 East, ColoradOS/2

Ad Close Date / July 6

Introducing...
the **OS/2 DEVELOPER**
Lotus
NOTES
Products and Services Guide

Each issue of *OS/2 DEVELOPER* will now feature this special advertising section...devoted exclusively to products and services relating to Lotus Notes.

Costs for an 1/8 page ad are from \$350-\$450
Costs for a 1/4 page are from \$600-\$900

Call Now to reserve space
in the September/October issue...
Theme: Database Development

Deadline is July 6

Call:

Kristin Morgan
212-626-2498

Jagre SmartLink and Lotus Notes: A Winning Combination!

File Open

File Save

E-mail

Fax

Pager

OCR

Audio

Video

iRetriever

Help

In today's computer world, communicating effectively and quickly could be the difference between your company being a dead company or being a quick company.

Using Lotus Notes within your company, gives your company the ammunition it needs to be a quick company.

Now, Jagre, Inc. provides your company with the competitive advantage to being -- and staying -- a quick company: **Jagre SmartLink**.

Jagre SmartLink is a set of Lotus Notes-enabled tools grouped together in one place to empower your IBM OS/2-based desktop applications to communicate effectively and quickly with Lotus Notes for Document Management, E-mail, Fax, Pager, OCR, Voice, Video, and User-based Agents to gets tasks done quickly.

With Jagre SmartLink, the powerful features of Lotus Notes are now a button click away!

Click on any Jagre SmartLink tool for instant access to Lotus Notes. Now, you can use Lotus Notes and the Lotus Notes Companion Products with your desktop applications to communicate quickly with someone down the hall or across the world!

Now, your ammunition arsenal is complete with Jagre SmartLink and their arsenals are not!!

Jagre SmartLink

For a free demonstration copy of Jagre SmartLink to discover how it can help your company be a quick company, not a dead company, call or write Jagre, Inc. at (617)424-8302, 102 Gainsborough St. #202E Boston, MA 02115.

Circle Reader Service Number 34

Jagre SmartLink

★ Products & Services Guide Continued

Object Technology for OS/2 Developers

Make DB Technologies, Inc. your first stop for powerful, object-oriented tools for today's popular development environments. Our current product line includes:

OS/2 Custom Controls
Calendar
Entry Field Object Pack
Digital Clock
Progress Indicator

VisualAge Parts
Calculator Parts
Calendar
Digital Clock
Progress Indicator

VX-REXX Objects

Digital Clock
Enhanced Picture Box
Entry Field Object Pack
Line Object

Multimedia Object Pack
Progress Indicator
VX-CALENDAR

We also provide consulting and custom development services. Our **Object Technology Group** provides custom software solutions for customers with unique requirements. If you need custom development, give us a call.

DB Technologies, Inc.

2420 Briar Oak Circle
Sarasota, Florida 34232

Sales: (800) 830-8703, (813) 379-8766
Object Technology Group: (813) 378-3760
Fax: (813) 377-7392
CompuServe: 72123,3661
Internet: 72123.3661@compuserve.com

Circle Reader Service Number 33

ORBitize™ v1.1' **Build and Browse IDL with ORBitize™**

ORBitize™ eases the development of distributed objects using an object request broker (ORB). It provides an intuitive graphical user interface for creating and browsing OMG IDL definitions rapidly. ORBitize™ browsing capabilities let you easily view and understand IDL definition hierarchies and quickly navigate between definitions. ORBitize™ eliminates the need to learn IDL syntax, allowing new developers to quickly produce error-free IDL.

- ORBs: SOM, ObjectBroker, Orbix, TME, DOE, ORBeline
- Platforms: OS/2, MS-Windows, NT, UNIX

74 Northeastern Blvd., Ste. 8B,
Nashua, NH 03062
v: 603-594-0525 f: 603-594-0527
email: info@netlinks.com

NetLinks
Technology, Inc.

Circle Reader Service Number 32

Products & Services Guide Ads

are read by 36,000+
professional OS/2 developers
and

The Cost Can't Be Beat!

(1/8 pg. ads as low as \$364/insertion)

Call Today!

Kristin Morgan / 212-626-2498



This article is excerpted with permission from Lotus Notes Release 3 in the OS/2 Environment, John Wiley & Sons, ISBN 0-471-13127-X. **By TONY WALSH**

A Lotus Notes REXX API

IBM used the Lotus Notes C API Toolkit to produce a REXX interface for Notes OS/2 Workstations and Servers. The product is called InterfloX. 1-2-3 for OS/2 and Ami Pro for OS/2 also have built-in REXX interfaces. Single REXX scripts may be written and executed from either 1-2-3 or Ami Pro that contain calls to the Notes REXX API and either the 1-2-3 or Ami Pro REXX APIs, respectively. These scripts may be executed interactively or from 1-2-3 or Ami Pro macros.

This is a two-way interface; that is, 1-2-3 and Ami Pro can both read from and write to Notes using REXX. The NotesSQL driver used in the Lotus Suite for Windows gives a similar capability by allowing Notes data to be accessed from 1-2-3 and Approach as an ODBC data source. This interface is only one-way. It does not allow 1-2-3 or Approach for Windows to write to Notes, nor does it provide a high-level programming environment underlying the Suite, as REXX does. Lotus Suite Windows users will have to wait until LotusScript is available for Notes, 1-2-3, and Ami Pro to match the capabilities REXX and these APIs provide!

OVERVIEW—INTERFLOX

InterfloX provides the following features:

- Summarizes the 200-plus low-level Notes access functions in the Lotus Notes C API into 40-plus high-level REXX functions. Though the REXX API has only a subset of the functionality of the C API, the functions it provides meet the programming requirements of many common Notes API applications.
- Accepts REXX scripts as input when installed on Notes OS/2 Workstations and Servers. InterfloX-defined REXX functions in scripts are processed into calls to Notes DLLs by the InterfloX application RXL-

NOTES.DLL, which was created using the Notes C API Toolkit.

- Builds OS/2 applications that may be distributed to Notes OS/2 workstations and servers.
- Does not require the C API to be installed on the workstations and servers on which its applications are executed. Notes must be installed on the workstations and servers on which REXX API applications are to be executed. Only OS/2, Notes, and InterfloX are required for Notes API REXX scripts to run.

FUNCTIONALITY

The REXX API can be used to:

- Create, delete, and make copies of Notes databases
- Read, create, and edit database documents
- Read, create, and edit document fields, including rich text fields
- Search for databases and documents
- Search Notes views
- Send Notes mail
- Read user and environmental information
- Perform enhanced error checking and data validation.

The above list is shorter than the equivalent list in the section of this chapter on the C API. The REXX API has the same Notes limitations as the C API. Additionally, it cannot:

- Create forms or views
- Work with encrypted data
- Read unread marks.

EXECUTION

A REXX API application may be executed at Notes OS/2 workstations:

- At an OS/2 Command prompt by typing the filename of the application. For example, typing DOWNLOAD at an OS/2 Command prompt would execute the REXX script DOWNLOAD.CMD.

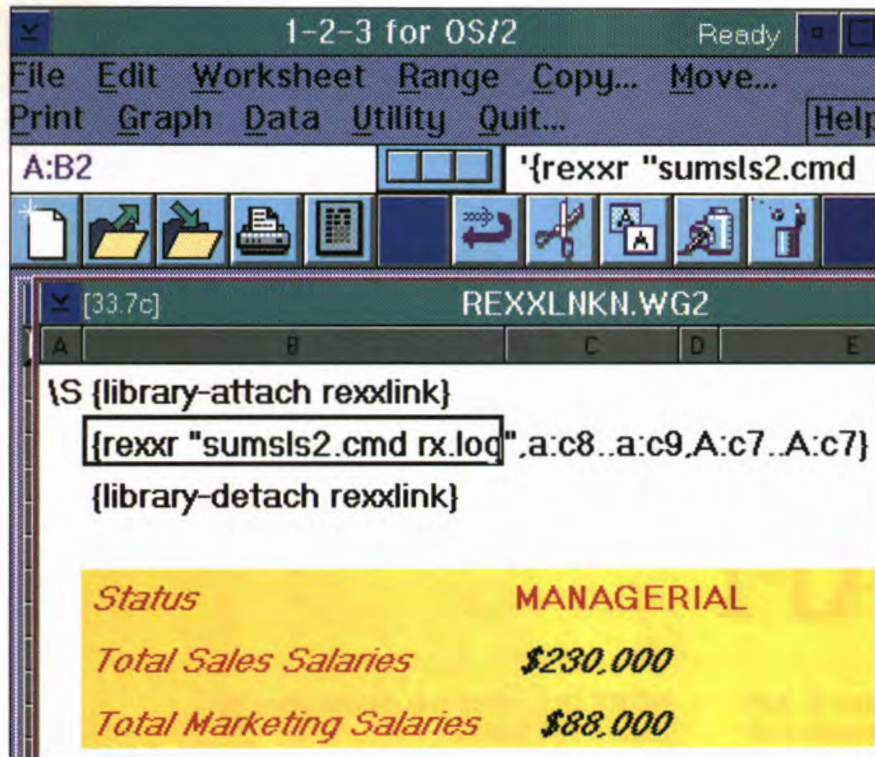


Figure 1. 1-2-3 for OS/2.

```

/*****
/*****SUMSLS2.CMD - Returns summary Notes data to 1-2-3**/
/*****

/*****
/*****The Notes database containing the employee data*****/
/*****
db_name='SALES.NSF'

/*****
/*****Initialize the REXX API*****/
/*****

If rxfuncquery('LTSNOTES')>0 Then Do
  rc=rxfuncadd('LTSNOTES', 'RXLNKN', 'LTSNOTES')
end

/*****
/*First NOTES function must initialize the Notes Environment */
/*****
call LTSNOTES 'NOTESINIT'
say "got back from init" result
if result>0 then signal ErrExit

/*****
/****Open the Sales database****/

/*****
call LTSNOTES 'DatabaseOpen',db_name,'db_id'
if result>0 then do
  say "Error opening database" db_name
  signal ErrExit
end

/*****
/***** Validate args inRange and status as ranges *****/
/* the 1-2-3 REXX interface uses RANGE1 & RANGE2 as the ***/
/***** returned range labels *****/
/*****
parse ARG inRange, status

```

Figure 2. Listing of REXX program that returns summary Lotus Notes (continued on page 55.).

- From the Workplace Shell, using a program icon
- From a Notes SmartIcon
- From a Button on a Notes form
- As background macros, scheduled in the Workstation's local Name and Address Book.

ADVANTAGES

- Easy installation. No compilers need to be installed. REXX is OS/2's native scripting language. The OS/2 system editor is ideal for creating REXX API applications.
- No compilation required. REXX scripts can be executed straight from the editor.
- Rapid application development. REXX is a high-level language, and applications can be developed, tested, and debugged much more quickly than in C.
- Minimal disk space requirements. The REXX API only uses 3 MB of additional disk space. Scripts are in ASCII format.
- Ideal for prototyping applications. Applications usually go through a series of design, test, and modification cycles before the application is ready for production. This process is much faster with REXX, as any OS/2 PC can be used to modify the application, and no compilation is required. Once the application design is final, all, part, or none of it may be recoded in C for enhanced functionality or performance.

DISADVANTAGES

- Less functionality than the C Toolkit. Some applications require features that are not available in the REXX API.
- Performance. REXX programs do not execute as quickly as C programs. However, the difference in performance may be either minimal or inconsequential if:
 - The REXX script or C program is primarily a wrapper for calls to Notes DLLs, which are doing the bulk of the processing
 - The script or C program is scheduled to execute at night on the Notes Server, where it may make little difference if the program takes 5 or 25 minutes to run
 - The script or C program performs little or no screen I/O. Screen I/O is the area of greatest difference in performance between C and REXX. Thus, a typical API application that is scheduled to run in the middle of the night to perform maintenance

tasks against a Notes database and that does no screen I/O may meet all the application's requirements written in REXX rather than in C, with a considerably reduced amount of development and maintenance effort.

- User-modifiable code. One disadvantage of interpreted languages like REXX is that users running a REXX script can modify the script code. Compiled code can be modified only if the users have access to the source code and a compiler. This is an issue only for REXX scripts that are run interactively by users on Notes Workstations. Most REXX scripts run on the Notes Server and can be neither viewed nor modified by Notes Workstation users unless they can also access the Notes Server as a file server. Notes users should never be allowed to access Notes Servers as file servers since this gives them the capability to open Server databases as "Local" databases, completely circumventing ACL security.

1-2-3 FOR OS/2—REXX INTEGRATION

1-2-3 for OS/2 includes a REXX interface that allows 1-2-3 macro programs to execute REXX scripts, set REXX variables with values from 1-2-3 cells, and set 1-2-3 cells with values from REXX variables. As Notes REXX API programs can also read and write REXX variables, users with 1-2-3 for OS/2, Notes for OS/2, and the Notes REXX API installed on their PCs can run 1-2-3 applications that use REXX scripts to pass data between 1-2-3 cells and Notes fields.

SAMPLE APPLICATION

The 1-2-3 spreadsheet illustrated in Figure 1 uses the Notes REXX API to summarize Notes data grouped on field data stored in a 1-2-3 cell.

The 1-2-3 macro \S, stored in cells A:B1..A:B3, consists of three lines:

```
{library-attach rexxlink}
```

The first line in the macro loads the 1-2-3- REXX DLL into memory.

```
{rexxr "sums1s2.cmd"
rx.log",a:c8..a:c9,a:c7..a:c7}
```

The second line runs the REXX script sums1s2.cmd, passing two 1-2-3 ranges as arguments, logging any screen output from REXX to the file rx.log. The first of the ranges passed as arguments, a:c8..a:c9, contains the string MANAGERIAL,

```
IF inRange <> "RANGE1" THEN DO
  CALL Display123Error "First Arg to SUMSLS2.CMD not a range."
  RETURN
END

PARSE VAR range1.0 cells sheets cols rows
IF sheets > 1 ] cols > 1 THEN DO
/* quit if too many sheets or cols */
  CALL Display123Error "range1 must lie in 1 column on 1 sheet."
  RETURN
END

IF status <> "RANGE2" THEN DO
  CALL Display123Error "2nd Arg to SUMSLS2.CMD not a range."
  RETURN
END

PARSE VAR range2.0 cells2 sheets2 cols2 rows2
IF sheets2 > 1 ] cols2 > 1 THEN DO

/* quit if too many sheets or cols */

  CALL Display123Error "range2 must lie in 1 column on 1 sheet."

  RETURN

END

/*****
/* Get all documents in the database - DATA and NONDATA*****/
/*****
call ltsnotes 'DatabaseSearch',db_ID,'docid_list'
if result<>0 then signal ErrExit

/*****
/***** Initialize summary variables *****/
/*****
TotSalS = 0
TotSalM = 0

/*****
/*****Set REXX variable RXStatus to cell A:A7*****/
/*****The Get123Cell 1-2-3 REXX function takes the ****/
/*****form: ****/
/**Get123Cell(Range, sheet, column, row) where: **/
/**Range is the range containing the cell whose **/
/**value will set the REXX variable; **/
/**sheet is the integer displacement value of the **/
/**sheet relative to Range. That is, if Range is **/
/**on sheet A, say, and sheet is 2, then the **/
/**desired cell is on sheet B; **/
/**column and row are the relative displacement **/
/**values for row and column respectively, In this **/
/**case, as 1 and 1, they point to the first column **/
/**and row in the range. **/
/*****
RXStatus = Get123Cell(status, 1,1,1)

/*****
/* If the DatabaseSearch function returned a valid whole */
/* number, read and process documents. The RXStatus */
/* field is set by 1-2-3 to limit the query to either */
/* MANAGERIAL or NONMANAGERIAL employees: the */
/* query accumulates Salaries for the chosen group */
/* of employees who are broken out into "Sales" and */
/* "Marketing" according to the value of the Department */
/* field stored in the REXX stem as doc_info.2.4.3 */
/* doc_info.2.6.3 contains employee salary, and */
/* doc_info.2.5.3 contains the employee status. */
/*****
```

Figure 2. Listing of REXX program that returns summary Lotus Notes (continued on page 56).


```

if datatype(docid_list.0,'N') then
  do i=1 to docid_list.0
    doc_info=""
    /* Get all the available info about the document */
    call ltsnotes 'DocRead',db_ID,docid_list.i,"doc_info"
    if result<>0 then signal ErrExit
    if doc_info.2.5.3=RXStatus then do
      If doc_info.2.4.3 = "Sales" then
        TotSalS = TotSalS + doc_info.2.6.3
      If doc_info.2.4.3 = "Marketing" then
        TotSalM = TotSalM + doc_info.2.6.3
    end
  end
end /* do */

/*****Pass the accumulated figures back into 1-2-3*****/
/*****Set123Cell "RANGE1", 1, 1, 1, "NUMBER", TotSalS
Set123Cell "RANGE1", 1, 1, 2, "NUMBER", TotSalM

/****Close the Notes database.****/
call LTSNotes 'DatabaseClose',db_id
if result<>0 then do
  say "Error closing database" db_name
  signal ErrExit
end

/*****Close the Notes Environment. *****/
call ltsnotes 'notesterm'
if result<>0 then call showerror

exit

/* ErrExit - Exit point for error conditions. Display error info and */
/* terminate the Notes Environment (NotesTerm) before exiting*/
ErrExit:

/* Print out any error messages */
call showerror

call ltsnotes 'notesterm'
if result<>0 then call showerror

exit

/*showerror - Display error information contained in the stem */
/* LTSNOTES_LAST_ERROR. Also reset message count to 0 */
/*****/showerror:
if(datatype(ltsnotes_last_error.0,'W')=1) then
  do
    do i=1 to ltsnotes_last_error.0
      say "Error:" ltsnotes_last_error.i
    end
    ltsnotes_last_error.0=0
  end
end

return

```

Figure 2. Listing of REXX program that returns summary Lotus Notes (continued from page 55).

used as the group field value for the summary operation in the REXX script. The REXX script summarizes the Salary field in the Notes database SALES.NSF. Only MANAGERIAL salaries from the Sales and Marketing departments are summarized. The second range, a:c7..a:c7, points at the spreadsheet range where the summary data from REXX will be returned.

{library-detach rexxlink}

The final line unloads the 1-2-3 REXX DLL from memory. Figure 2 contains the REXX script SUMSLS2.CMD and is liberally commented.

Further information on Lotus OS/2 products is available in *Lotus Notes Release 3 in the OS/2 Environment*. This book not only contains working examples on the usage of InterfloX to integrate Notes and the Lotus OS/2 Suite but also has detailed information on Notes architecture and its integration into the wider IBM environment, including chapters on Notes OS/2 protocols and Notes integration with DB2.

Tony Walsh taught math at Brixton College, London, England, from 1974 to 1984, was a COBOL programmer at Data General, Westboro, Mass., from 1984 to 1986, and was a Systems Engineer for Information Builders Inc. from 1986 to 1987. He has been at Lotus, Cambridge, Mass., since 1987 and is currently a Senior Communications Consultant in the Integrator Alliances group. Other titles held at Lotus include Senior Software Engineer, Field Support Services Consultant, and Product Manager for 1-2-3 for OS/2, 1-2-3/M, and 1-2-3 for VAX/VMS.

InterfloX can be ordered by calling 1-800-IBM-CALL, asking for department S47, and requesting product number 5764-053.



The staff of OS/2 Developer put together this special section to guide you through the various software tools specifically for object-oriented development. The products in this guide are based on surveys sent to vendors; they are provided as a service to you and are free to vendors. These listings do not represent an endorsement by OS/2 Developer. Although every effort has been made to ensure accuracy, we do not assume any responsibility for error or omission in this section.

Object-Oriented Tools Buyers Guide

ATHENA DESIGN INC.

Circle No. 101

Mesa 2.02 combines traditional spreadsheet technology with objects and 32-bit multi-threading to take advantage of OS/2. The latest upgrade, free to all registered users, includes Excel import and export, DDE links, and Zoom, as well as other enhancements and bug fixes. Mesa features REXX as its native scripting language and adds multiple MScript commands. This list price includes 90 days of technical support. Price: \$199.

Athena Design Inc., 332 Congress St., Boston, Mass. 02210, (800) 315-MESA or (617) 426-6372, fax (617) 426-7665.

CADRE TECHNOLOGIES INC.

Circle No. 102

Teamwork for Structured Methods 6.0 products improve software quality and reduce development costs. Automating industry-proven, real-time structured analysis and information modeling methods, its tools facilitate the creation of specification and models, help ensure consistency of data, perform automated checking, and automatically generate SQL DDL from the information model. Teamwork for Structured Methods tools also provide printing options and support popular desktop publishing systems and formats. Price: contact vendor.

Cadre Technologies Inc., 222 Richmond St., Providence, R.I. 02903, (800) 743-2273 or (401) 351-5950, fax (401) 455-6800.

CLASSIC SOFTWARE INC.

Circle No. 103

Btrvgen++ 3.0 is a database designer and C++ code generator for Btrieve. Features include an integrated Btrieve DDF file editor and generated code, which supports 16- and 32-bit OS/2. All Btrieve data types are supported. In addition, a special bundle is avail-

able with the Btrv++ product for \$349. Price: \$229.

Btrv++ 3.0 is a C/C++ library interface to Btrieve. It implements all Btrieve operations as member functions as well as functions for SQL queries and file and index creation. This multicompile platform product supports 16- and 32-bit OS/2 and contains complete record and field-based access. Royalty-free. Price: \$249.

Classic Software Inc., 900 Victors Way, Ste. 170 East, Ann Arbor, Mich. 48108, (800) 677-2952 or (313) 913-8075, fax (313) 971-3287.

COMPILER RESOURCES INC.

Circle No. 104

Yacc++ and the Language Objects Library 2.0 is an object-oriented rewrite of lex and yacc that automatically generates C++ classes of lexer, parser, and AST (abstract syntax tree) objects from grammars. The library provides classes for various models of AST, error, input, lexer, parser, and symbol table semantics. Price: \$995 for a single-user license.

Compiler Resources Inc., 85 Main St., Ste. 310, Hopkinton, Mass. 07148, (508) 435-5016, fax (508) 435-4847.

DIGITALK INC.

Circle No. 105

Visual Smalltalk Enterprise 3.0.1 allows programmers to create mission-scalable applications with easily assembled components. This object-oriented, client/server tool supplies visual component assembly and the ability to create new components. In addition, Visual Smalltalk Enterprise offers version control, configuration management, and a multiuser environment. Price: \$5,000.

Digitalk Inc., 5 Hutton Centre Dr., Santa Ana, Calif. 92707, (800) 546-6400 or (714) 513-3000, fax (714) 513-3100.

EXPERSOFT CORP.**Circle No. 106**

XShell DOME 3.5 is a distributed object management environment including XShell Distributed ORB and integrated tools. XShell unifies the corporate network by transforming legacy, client/server, and object applications into distributed, interoperable objects that developers can use to connect legacy and client/server systems to create large-scale distributed environments. Price: \$9,600 per developer suite.

Expersoft Corp., 6620 Mesa Ridge Rd., San Diego, Calif. 92121, (800) 366-3054 or (619) 546-4100, fax (619) 450-0644.

FOUR SEASONS SOFTWARE**Circle No. 107**

4S-Report 2.0 is an information retrieval and analysis tool that offers users a cost-effective method of interrogating data and producing customized reports without programmer intervention. This product supports both character-based and GUI environments and can be used as an independent tool able to run simultaneously on a range of database packages, including Informix, Oracle, Sybase, Ingress, DB2/6000, HP/Allbase, RDB, Xbase, incore memory files, and flat ASCII files. Price: \$320.

SuperNOVA 5.0 is a 4GL application development environment featuring online application partitioning, visual application partitioning management, workgroup management, and configuration management of application development. This product allows developers to build all applications in one development environment and deploy the applications throughout their companies regardless of the target hardware, data manager, or network. Portable across multiple platforms, SuperNOVA supports character-based and GUI interfaces. In addition, it supports the following DBMSs and file interfaces: Informix, Oracle, Sybase, Ingres Teradata, DB2/6000, HP/Allbase, dBASE, COBOL, and more. Price: \$1,740.

Four Seasons Software, 2025 Lincoln Hwy., Edison, N.J. 08817, (800) 949-0110 or (908) 248-6667, fax (908) 248-6675.

IBM**Circle No. 108**

TeamConnection 1.0 for OS/2 integrates software configuration management functions for object-oriented (C++) 3 GL and 4 GL languages and object-based repository services on a semantic model for tool integration to support application development in an OS/2 client/server team programming

environment. TeamConnection supports the development and building of OS/2 and MVS client/server and distributed applications. Functions are available through GUI and command-line interface. Price: Under \$2,000.

IBM, P.O. Box 12195, RTP, N.C. 27709, (800) 426-2279 or (919) 254-4760.

IBM**Circle No. 109**

VisualAge C++ 3.0 for OS/2 combines a full visual programming environment with a set of IBM Open Class Libraries and a 32-bit C++ compiler. Enhancements include an integrated compiler, debug and browse support for building SOM objects directly from C++, and an interactive C++ browser. Price: contact vendor.

IBM, 1150 Eglinton Ave. East, North York, Ont., Canada M3C 1H7, (800) IBM-3333.

INTELLIGENT OBJECTS CORP.**Circle No. 110**

SQL Objects 2.0.1, a database class library, supports multiple operating systems, including OS/2. In addition, it supports 22 different databases natively and over 74 with ODBC, such as DB2/2, Sybase, Oracle, SQL Server, Watcom SQL, Gupta SQL Base, Informix, ODBC, Netware SQL, and Btrieve. Price: \$699.

Intelligent Reports 1.0 is a database reporting and access product for OS/2 that supports over 22 databases by incorporating SQL Objects. Intelligent Reports supports multithreading, an easy-to-use interface, and multiple report styles such as cross tabular, columnar, labels, and forms. Price: \$499.

Intelligent Objects Corp., 47 Stonewall St., Cartersville, Ga. 30120, (800) 876-6585 or (404) 382-6585, fax (404) 382-6374.

INTERACTIVE SOFTWARE ENGINEERING INC.**Circle No. 111**

ISE Eiffel 3.3, an object-oriented development environment, includes a combination of interpretation for rapid application development and compilation for optimized executables visual workbench; numerical, database interface, GUI libraries; analysis/design workbench; multiplatform GUI-builder; and C-code generation. Price: \$995.

Interactive Software Engineering Inc., 270 Storke Rd., Ste. 7, Goleta, Calif. 93117, (805) 685-1006, fax (805) 685-4976.

JBA INTERNATIONAL**Circle No. 112**

Guidelines 3.1 is an object-oriented OS/2-hosted GUI development environment that automates most of the

creation of a GUI application. The developer is able to create applications that range from fully event-driven procedural to event-driven object-oriented with SOM and DSOM support. Guidelines creates OS/2 and Windows 3.1 applications supported on DB2/2, DB2/400, and ODBC SQL servers. Price: \$595 to \$8,000 depending on options.

JBA International, 3701 Algonquin Rd., Crossroads Ctr., Ste. 1000, Rolling Meadows, Ill. 60008, (800) JBA-INTL or (708) 590-0299, fax (708) 590-0394.

MICADO SOFTWARE CONSULT GMBH**Circle No. 113**

micWOP (micado Workplace for Visual Smalltalk/V Corporate) is a development environment for Visual Smalltalk, OS/2 2.x 32-bit, and/or Win32 that includes and uses several commercially available Visual Smalltalk third-party products and add-ons developed by micado. These include a translator, a documentation manager, an environment manager, a help manager, a session starter, and several other extensions. For Database access, it uses the ODBC driver, which provides access to most of the commercially available databases such as MS SQL Server, Sysbase SQL Server, Oracle version 6+7, DB2/2, DB2-IBMSQL/400-IBMSQL/DS via Gtway, Gupta, dBase, II+IV, Novell SQL+Btrieve, and Excel. Price: starts at \$4,995.

micado FrameWorks supplies basic functionalities for development projects and includes three components: micado Object FrameWork, micado Persistence FrameWork, and micado Application FrameWork. The micado Object FrameWork extends the Smalltalk basic functionality with transaction handling, typing, and relationships, while the micado persistence FrameWork extends the Smalltalk basic functionality with mechanisms to store Smalltalk objects on different storage media. Smalltalk objects can be used without abusing the syntax and semantic of a Smalltalk environment. The micado Application FrameWork extends the Smalltalk basic functionality with abstract object controls, base models, and business models. Price: contact vendor.

micado Software Consulting GmbH, Reutherstraße 1, 53773 Hennef, Germany, 22-42-87-14-50, fax 22-42-87-14-55.

MICRO FOCUS**Circle No. 114**

Object COBOL 3.3 is a 32-bit COBOL compiler supporting both COBOL and object-oriented COBOL programming

on OS/2. It is a 32-bit product for migrating existing and developing new COBOL applications that take full advantage of today's PCs and operating systems. Price: \$1,500.

COBOL Workbench 3.3 is a comprehensive, integrated suite of programmer productivity tools designed to facilitate the development and maintenance of high-quality, industrial-strength OS/2 applications. Workbench is a 32-bit, object-oriented programming environment that provides a 32-bit compiler and supports both COBOL and object-oriented COBOL programming. Price: \$2,500.

Micro Focus, 2465 East Bayshore Rd., Palo Alto, Calif. 94303, (800) 872-6265 or (415) 856-4161, fax (415) 856-6134.

MULTIQUEST CORP. Circle No. 115

S-CASE 2.0 is an object-oriented analysis and design tool that uses Booch notation to graphically illustrate and model software systems. High-quality C++ code can be generated directly from the models. S-CASE allows iteration through the design, code, and test cycle while keeping the models and code in sync. Price: \$495.

MultiQuest Corp., 1931 North Mechem Rd., Ste. 318, Schaumburg, Ill. 60173, (708) 397-9930, fax (708) 397-9931.

NETWORK TECHNOLOGY CORP. Circle No. 116

MEDIAscript OS/2 Desktop 2.5 provides capture, edit, and authoring tools for the design of multimedia applications using audio, video, images, and animation in OS/2. It employs icons in a visual design environment and fully supports both ActionMedia II and MPEG digital video technology. Price: \$895.

Special Edition is identical to the Desktop Edition in its object-oriented visual design environment; however, it does not include high-performance video drivers for ActionMedia II. It fully supports MPEG Video and software video through MMPM/2 and includes enabling technology for other OS/2 applications. Price: \$395.

Network Technology Corp., P.O. Box 240, Dobbs Ferry, N.Y. 10522, (914) 478-4500, fax (914) 478-4507.

OBJECTS/FX CORP. Circle No. 117
Visual Companion Object Developer's Kit 1.0 is a framework of Smalltalk classes for projects requiring spatial visualization and analysis. Developers can directly use the software capabilities of

Visual Companion such as map, table, and picture viewing; spatial analysis; thematic displays; and communication with other applications. This kit can be used to enhance existing Smalltalk applications or to rapidly develop new "map-centric" applications. Price: \$5,000.

Visual Companion 2.0 is a geographic visualization and analysis product offering map, table, and picture viewing; spatial analysis; and thematic displays. Visual Companion is also designed to be embedded within other applications. By using the messaging protocol, any application can issue commands that manage map windows, highlight map objects, display thematic maps, and retrieve the results of spatial queries performed within the intuitive map interface. Price: \$995.

Object Mapper 1.0 is a tool for building geographic object databases for use in Object/FX products. Map objects may be created by hand or imported automatically from various geographic data sources or street address files. Object Mapper also supports scanned map backgrounds, multiple zoom levels, configurable display options, and the distribution of customized databases across a client server enterprise. Price: \$1,500.

Objects/FX Corp., 2515 Wabash Ave., Ste. 100, St. Paul, Minn. 55114, (612) 644-6064, fax (612) 644-0366.

OBJECTS INC. Circle No. 118

Layout 3.0 is an object-oriented application development tool for OS/2 that allows both programmers and nonprogrammers to graphically build applications without using a computer language. Price: \$299.95.

Objects Inc., 99 Rosewood Dr., Danvers, Mass. 01923, (800) 424-6644 or (508) 777-2800, fax (508) 777-0180.

OBJECTSPACE INC. Circle No. 119

ObjectCatalog 1.0 is a cross-platform component reuse facility that helps developers reuse classes, patterns, designs, documentation, and other components in a local or highly distributed development environment. Components are classified and located based on descriptions entered into a catalog. The GUI interface and classification method facilitates the searching process. Price: \$795 per client.

STL <ToolKit> 1.0, ObjectSpace's implementation of the ANSI Standard, is a framework of containers, iterators, and algorithms. In addition to including standard collections such as vectors, lists, deques, sets, amps, stacks, and queues, STL <ToolKit> includes

support for multithread development. Price: \$149.

Systems <Toolkit> 1.0 is a comprehensive set of cross-platform C++ foundation classes. In addition to the inclusion of a multithread-enabled implementation of STL, System <Toolkit> interfaces with the operating system features such as files, sockets, threads, and processes. Price: \$375.

ObjectSpace Inc., 14881 Quorum Dr., Ste. 400, Dallas, Tex. 75240, (800) OBJECT-1 or (214) 934-2496, fax (214) 663-3959.

OBJECT TECHNOLOGY INC. Circle No. 120

ENVY/Developer is a bundle of four separate tools. These tools enable the development, release, and maintenance of applications written in Smalltalk. All source and object code is maintained online in a shared repository. This multiuser architecture provides version control and configuration management, which are integrated with Smalltalk's object-oriented programming environment. ENVY/Developer is available in versions 1.43 for VisualWorks and 1.44 for Visual Smalltalk. Price: \$3,000.

Object Technology Inc., 301 East Bethany Home Rd., Ste. A-100, Phoenix, Ariz. 85012, (602) 222-9519, fax (602) 222-8503.

PARCPLACE SYSTEMS INC. Circle No. 121

VisualWorks 2.0 is a client/server tool for building portable applications using object-oriented technology. A database application creator is included for rapid application development. Applications are instantly portable across multiple platforms, are scalable across the enterprise, and can have their functionality distributed between both clients and servers. Price: \$2,995.

ParcPlace Systems Inc., 999 East Arques Ave., Sunnyvale, Calif. 94086, (800) 759-7272 or (408) 481-9090, fax (408) 481-9095.

POET SOFTWARE CORP. Circle No. 122

Poet 3.0 is a C++ Object DBMS that supports encapsulation, inheritance, and polymorphism. Available for 14 different operating systems, it comes as a single-user or a client/server version. Featuring schema versioning, check-in/check-out, and online backup, Poet also supports Developer Workbench, which includes class and object browsers. Price: \$499 for single-user and \$3,999 for client/server SDK.

Poet Software Corp., 999 Maker Way, Ste. 100, San Mateo, Calif. 94404,





(800) 950-8845 or (415) 286-4640, fax (415) 286-4630.

**POPKIN SOFTWARE
SYSTEMS INC.**

Circle No. 123

System Architect 3.0 is a repository-based application development tool that supports the analysis and design of client/server, MIS, and real-time applications. The tool supports multiple methodologies in four approaches to application development: structured planning, analysis, and design; object-oriented modeling; data modeling; and business process reengineering. In addition, System Architect features a shared, customizable repository with generation facilities and an open API. Price: \$1,395.

Popkin Software Systems Inc., 11 Park Place, New York, N.Y. 10007, (800) 732-5227 or (212) 571-3434, fax (212) 571-3436.

PROSA SOFTWARE

Circle No. 124

Prosa/om Object Modeling Environment 4.0 covers development from identifying and qualifying classes to specification, design, and implementation with OMT methods. Prosa/om has easy-to-use graphic editors, documentation tools, and code generators. In addition, the checking between models is automatic and interactive. Documentation is prepared concurrently with development using Prosacdm Concurrent Document Manager. The Prosac++ tool generates complete C++ and C

codes from OMT models; the Prosasql tool generates SQL code. Price: \$3,560.

Prosa Software, P.O. Box 9, FIN-90101 Oulu, Finland. 358-81-376-128, fax 358-81-371-754.

**TECHNICALLY
SPEAKING INC.**

Circle No. 125

Show N Tel 2.0 is an application builder for voice-processing, corporate fax, messaging, call center, and other CTI solutions. Included are over 300 pre-programmed building blocks, called PowerBlocks, that reduce development time. An open architecture allows complete customization, unlimited scalability, and links to virtually any database and host system. Price: \$995 to \$2,995.

Technically Speaking Inc., 333 Turnpike Rd., Southboro, Mass. 01772, (508) 229-7777, fax (508) 229-8777.

TOWER TECHNOLOGY

Circle No. 126

TowerEiffel System 1.4 is a complete object-oriented programming system that includes a high-performance Eiffel 3 compiler, a development environment, programming tools, and a base set of reusable software components. The TowerEiffel System supports multilanguage development and interoperability between Eiffel, C, and C++. Price: \$249 to \$1,295.

TowerEiffel Booch Components 1.4 is the Eiffel version of Booch Components. Reengineered to take full

advantage of Eiffel's capabilities, the Eiffel Booch Components are a set of Eiffel clusters that provide a complete collection of efficient and adaptable domain-independent data structures and algorithms. Price: \$199 to \$995.

Tower Technology Corp., 1501 West Koenig Ln., Austin, Tex. 78756, (800) 285-5124 or (512) 452-9455, fax (512) 452-1721.

VISIX SOFTWARE INC.

Circle No. 127

Galaxy Application Environment 2.5 is created for corporations developing cross-platform, distributed object-oriented applications that can scale and accommodate changing business needs. The DAS (distributed application services) architecture facilitates development and deployment of distributed applications. Price: \$9,600 for the C version and \$12,100 for the C++ version.

Visix Software Inc., 11440 Commerce Park Dr., Reston, Va. 22091, (800) 832-8668 or (703) 758-8230, fax (703) 758-0233.

ZINC SOFTWARE INC.

Circle No. 128

Zinc Application Framework 4.0 is an object-oriented, internationalized C++ class library and visual development tool. This product supports OS/2 and 13 different languages. Price: \$798.

Zinc Software Inc., 405 South 100 East, Pleasant Grove, Utah 84062, (800) 638-8665 or (801) 785-8900, fax (801) 785-8996.

Point and click your way to fast development of GUI client-server applications.

Do it with Guidelines, a second-generation object-oriented graphical workbench. Guidelines lets you create graphical client/server applications for OS/2 and Windows clients. A built-in prompter lets you simply point and click or drag and drop from the desktop using more than 30 presentation manager controls.

Guidelines utilizes JBA's high-level object-oriented language (JOT) to describe applications. After the application is designed, JOT automatically converts to C++ code, so you don't have to learn its complex syntax and rules. But if you already know C++ and want to use it directly, you can. The conversion takes place regardless of the platform intended.

Guidelines saves time and simplifies the creation of graphical, platform-independent applications. Its scalable architecture lets you create anything from stand-

alone PC applications to complex enterprise-wide business solutions. Guidelines supports DB2/2, DB2/400, DB2/6000, DDCS, ODBC, Q+E Lib, and SQL.

Call for a free demo pack: 1-800-JBA-INTL.

In Canada: (905) 940-2442

JBA International
161 Gaither Drive, Ste. 200
Mt. Laurel, NJ 08054





New Products for OS/2

AM Development Environment for OS/2. This OS/2 environment is designed for the development of scalable, multimodule applications. It simplifies complex programming requirements for SQL, transaction management, and legacy-system renovation. AM integrates data from diverse, distributed sources including DB2 and Sybase. In addition, AM links to corporate data in IMS, VSAM, and other existing databases. The full exploitation of OS/2 architecture simplifies development of 32-bit multitasking, multi-threading applications. The product requires a 386-class or higher IBM PS/2 or 100% compatible; 8MB RAM; PS/2 VGA display or 8514 display or compatible; and/or OS/2 2.0 or higher.

Intelligent Environments Circle No. 130
Phone: (617) 942-0299, Fax: (617) 944-7785

BackMaster. MSR Development Corp. announces backup support for the QIC (Quarter-Inch Cartridge) tape drives that offer capacities up to 850MB. BackMaster enables Jumbo and Trakker 250 drives to store up to 350MB with DC2120XL tapes. In addition to the easy-to-use interface, this product provides OS/2 Warp compatibility, bidirectional OS/2 DOS/Windows data interchange via QIC minicartridge tapes, enhanced "boot from floppy" restore utility for rapid restart/recovery, and support for high-speed tape-accelerator cards.

MSR Development Corp. Circle No. 131
Phone: (409) 564-1862, Fax: (409) 560-5868

C-Cover 3.0. This branch coverage analyzer for C and C++ increases testing productivity by quickly identifying untested control structures, functions, C++ classes, and subsystems. A new feature allows forcing execution flow through specified branches and provides testing error conditions that are otherwise too difficult to reproduce. C-Cover analyzes mul-

tithreaded code, DLLs, device drivers, and system-level code. It includes a capability for sorting, filtering, and summarizing large amounts of coverage information. C++ is fully supported, including inline functions, templates, and exception handling.

Bullseye Software Circle No. 132
Phone: (800) 278-4268, Fax: (206) 524-3575

Galaxy Application Environment 2.5. This release includes enhanced distributed development features, including a visual datatag editor, a datatag iterator, and distributed help services. Galaxy expands its support for shared libraries, thereby improving the performance of Galaxy's applications by reducing the size of its executables and system overhead. The new combination box visual object combines list and text functionality, each of which is replaceable and subclassable.

Visix Software Inc. Circle No. 133
Phone: (800) 832-8668 or (703) 758-8230, Fax: (703) 758-0233

Guidelines Desktop. Guidelines Desktop, a graphical, event-driven application development tool, uses an object-oriented high-level language, called JOT, with full point-and-click code prompting to aid software developers. As a result, this product enables developers to bypass the complex syntax and rules associated with C and C++. At the same time, more advanced developers, who wish to use C++ and native platform-specific APIs, can call on the language directly whenever required. Guidelines Desktop also supports the reuse of existing C modules such as resources and dialogs. Visual editing and a range of layout tools are provided as a standard to aid screen design and facilitate the development of professional applications.

JBA International Circle No. 134
Phone: (800) JBA-INTL, Fax: (609) 231-9874



HLPDK/PA 12.5c. This multiplatform help authoring tool supports the creation of help files for multiple environments from one source file. It includes converter programs from WinHelp, the Norton Guides, and PopHelp to HLPDK. With these converters, help files that were created on DOS and Windows environments can be translated to OS/2 and the World Wide Web.

HyperAct Inc. Circle No. 135
Phone: (319) 351-8413, Fax: (319) 351-8413

OpenHASP. This hardware-based protection system prevents unauthorized access and execution of software. It is designed for software developers creating applications for Open Systems workstations and ultimately protects their programs from piracy and illegal use. OpenHASP features an advanced microprocessor chip that contains an electronic algorithm that is recognized by the protected software. Because it is algorithm-dependent, it cannot be disabled by automatic cracking procedures. Developers have access to OpenHASP's source code and can blend the OpenHASP protection routine into their software. The 84 bytes of nonvolatile memory allows developers to custom-program individual keys for purposes such as user identification and multiple program authorization.

Aladdin Knowledge Systems Ltd. Circle No. 136
Phone: (800) 223-4277 or (212) 564-5678, Fax: (212) 564-3377

Revolve 3.0. With this visual analysis tool for legacy COBOL software systems, software developers can interactively examine and inventory existing applications. Revolve provides an enterprise view of applications and analyzes the relationships between the elements that link the data together, including moves, compares, calls, and files. The information is presented in hierarchical, collapsible list boxes that allow users to view text panels of data selected. The enhanced layout provides a more consistent, well-structured presentation of data. Revolve's administrative functionality enables a single person to manage projects for a large group.

Micro Focus Circle No. 137
Phone: (415) 856-4161, Fax: (415) 856-6134

RPMark Benchmark. RPMark Benchmark for client/server solutions tests all aspects of the client/server environment. As a result, RPMark's score reflects the impact of a single component, such as server, network application, drivers, or database software, on the entire environment, called the reference platform. Reference platform components will remain the same for one year and are based on popular products in use at client/server deployment sites. Benchmark includes composite suites of tests for decision support queries, OLTP (online transaction processing), and file serving applications. RPMark Benchmark is composed of three workloads that run concurrently: a transaction-based component, a decision support component, and a file serving/office automation services component.

Client/Server Labs Circle No. 138
Phone: (404) 552-3645, Fax: (404) 993-4667

SmartHeap 2.2. Now providing support for shared memory pools, SmartHeap, a memory management library and debugging toolkit, allows programmers to use ordinary heap operations on memory that is shared between processes. This product also expands and shrinks shared heaps on demand and automatically maintains a coherent view of memory between all processes that access the shared heap. SmartHeap provides improved performance and better memory utilization for applications with large dynamic memory requirements. Via a relinkable debug version of the library, SmartHeap detects and handles numerous memory errors, such as leakage, memory overwrites, and wild pointers.

MicroQuill Software Publishing Circle No. 139
Phone: (206) 525-8218, Fax: (206) 525-8309

TAWK Compiler 4.1. This compiler includes the new integrated TAWK Debugger, providing full-screen source-code-level interactive debugging so users can watch their TAWK programs execute. In addition, the TAWK Debugger features breakpoint, watched variables, stack trace, screen flip to view program output, and the ability to view or modify any program variable. Compiler is a general-purpose pro-

gramming language that includes pattern matching, searching, sorting, extensive text manipulation functions, built-in EMS/XMS memory management for large TAWK programs, and functions to pack and unpack binary record formats like database records.

Thompson Automation Software Circle No. 140
Phone: (800) 944-0139 or (503) 224-1639, Fax: (503) 224-3230

WindowBuilder Pro/V 2.0. Fully compatible with the new Visual Smalltalk 3.0 by Digitalk, this user interface building tool supports Visual Smalltalk's new features such as the event system and portability abstractions. This release can now also generate PARTS components in addition to Smalltalk code-generation techniques. The PARTS generated by WBPro/V can be placed and connected in the PARTS Workbench. This product also has the ability to generate ViewManager subclasses and application coordinator subclasses. WBPro/V includes a royalty-free license for distributing user applications, complete documentation, hypertext help, and 90 days of free technical support.

Objectshare Systems Inc. Circle No. 141
Phone: (408) 970-7280, Fax: (408) 970-7282

XDB-Workbench 4. This new version of the XDB-Workbench includes improved utilities, query performance, administration, and security, in addition to full support for DB2 3.1 SQL Syntax and all 600 buffer pools in DB2. The product provides programmers with a DB2 equivalent environment to develop, test, and execute complete DB2 applications on the desktop. The 4.0 database engine, COBOL Precompiler, and suite of development and administrative tools included are designed to help application developers produce mainframe-ready applications. These tools feature interactive facilities to create and alter tables, indexes, and locations; COBOL copy-book generation; full import and export capability; table editing and browsing; and an interactive SQL facility.

XDB Systems Inc. Circle No. 142
Phone: (800) 488-4948 or (410) 312-9300 ext. 420, Fax: (410) 312-9505

The staff of OS/2 Developer put together this special section to guide you through the various add-on tools and services for Notes applications for OS/2. The products in this guide are based on surveys sent to vendors; they are provided as a service to you and are free to vendors. These listings do not represent an endorsement by OS/2 Developer. Although every effort has been made to ensure accuracy, we do not assume any responsibility for error or omission in this section.

Take Notes....

APPLICATION

PARTNERS INC.

Circle No. 170

WIT The Notes Author 3.0 is a Notes server add-on tool that helps Notes developers build applications. This product facilitates prototyping and maintenance and eliminates development. Price: \$2,500.

Application Partners Inc., 100 Wood Ave. South, Iselin, N.J. 08830, (800) 603-7732 or (908) 549-5464, fax (908) 549-4435.

CAMBRIDGE

SOFTWARE GROUP INC.

Circle No. 171

The Formula Editor for Lotus Notes 1.0C is a power editor designed for Lotus Notes. It contains file save and retrieve, search and replace, printing, syntax checking, and CSG's proprietary formula formatter. Price: \$99.95.

Application Outliner for Lotus Notes 1.1B is an integrated add-on tool that organizes design elements into a structured Notes database. It helps debug and QA text applications, creates a data dictionary, and delivers user-quality documentation. Price: \$99.95; \$189.95 for the bundle of both products.

Cambridge Software Group Inc., 101 Rogers St., Ste. 406, Cambridge, Mass. 02142, (800) 792-7248 or (617) 577-0600, fax (617) 577-0602.

INFORMATION BUILDERS

Circle No. 172

EDA/SQL Copy Manager for Lotus Notes 1.02 combines data access technology with the Notes integration technology of Notrix from Percussion Software. The product makes it

easier to copy sales information, customer profiles, personnel data, financial results, and so on directly into the Lotus Notes database. Price: \$10,000.

Information Builders, 1250 Broadway, New York, N.Y. 10001, (212) 736-4433.

MICROAGE

INFOSYSTEMS SERVICES

Circle No. 173

Plan IT Works 1.0 is an application development suite used to assist project managers and developers in exploiting the full potential of their existing client/server tools. Plan IT Works guides development teams through a development method that promotes the rapid prototyping process.

The suite contains FrameWorks 1.0, a standards database that allows organizations to define the standards to be used during the design, development, and deployment of computer software systems. It also comes with ObjectWorks 1.0, which stores objects and routines in one central database for access by workgroups; DesignWorks 1.0, which provides the system designer with tools for documenting the detailed design information required to build an application; ProjectWorks 1.0, a high-end productivity tool that allows an organization to successfully track critical projects and their related activities; SupportWorks 1.0, a help desk application that is used by a central support group to provide online help desk support to internal and external clients; and TimeWorks 1.0, which is used to track both billable and nonbillable service hours provided to your clients.

MicroAge Infosystems Services,
37 Skyline Dr., Ste. 4304, Lake Mary,
Fla. 32746, (407) 333-2943, fax (407)
333-3905.

PERCUSSION SOFTWARE *Circle No. 174*
Notrix 2.0, a 4GL for Lotus Notes OS/2
servers, allows complex manipulation
of Lotus Notes data without Notes API
or C programming. Using REXX, you
quickly develop server add-in applica-
tions. Notrix allows you to store pro-
gram documents in a Notes database
and schedule them to run automati-
cally on the Notes server via the Notes
Name and Address Book. Price: \$1,495.

Percussion Software, 92 Montvale
Ave., Ste. 2100, Stoneham, Mass.
02180, (800) 283-0800 or (617) 438-9900,
fax (617) 438-9955.

SOFTBRIDGE INC. *Circle No. 175*
Automated Test Facility 3.5 tests multi-
user and distributed OS/2 applica-
tions. ATF's two-tiered architecture
supports client/server testing. From
one central point, ATF can run interde-
pendent, simultaneous, unattended
tests on multiple PCs, letting ATF test
realistic workflows, like those created
with Lotus Notes. Price: \$18,000 for the
base system.

Softbridge Inc., 125 Cambridge
Park Dr., Cambridge, Mass. 02140,
(800) 955-9190 or (617) 576-2257, fax
(617) 864-7747.

WINCHESTER
BUSINESS SYSTEMS INC. *Circle No. 176*
Lotus Notes Applications for Sales
Management provides the prototype for
sales automation, skills inventory, and
engineering change control. All of
these are in Lotus Notes. Price: \$3,995.

Winchester Business Systems Inc.,
138 Forest St., Winchester, Mass.
01890, (617) 729-7150, fax (617) 729-
7211.

PREDATOR/2
Release 2.1
Now available

THE EDITOR FOR UNTAMED PROGRAMMERS

PREDATOR/2 Control Panel

File Edit Search Buffers Tools Options Window Help

Buffer name File name MOD Lock RO

edhelp.sp c:\winword2\help\edhelp.sp

gament.chl c:\gameport

form.cpp c:\form.cpp

File Edit Search Buffers Tools Options Window Help

c:\gameport

extern char* format(const char* format ...)

static char buffer[512];

char* s = new char[1024];

va_list marker;

/* Initialize variable arguments */

va_start(marker, format);

vsprintf(s, format, marker);

if (strlen(s) < 51)

strcpy(buffer, s);

else

strcpy(buffer, "Output string too large");

Print the current buffer or marked block: (RWS) Line 60 of 75: (Col 16) Mon May 20 9:45 am

(RWS) MOD Line 792 of 1640: (Col 13) Mon Mar 20 9:45 am

For untamed OS/2 programmers

Your source code has never fallen into the grips of a more powerful editor. PREDATOR/2 is for developers who don't have the time to deal with rigid programming environments, who must change facilities to fit their exact requirements and who need to dramatically extend editing functions to attain peak productivity. With this editor, you'll work with the speed and strength of a true predator.

HIGHLIGHTS:

- Color syntax highlighting
- Single, multiple and detached window modes
- BRIEF, VI, EMACS, ISPF, CUA emulation
- Hypertext C/C++ browsing
- Built-in compiler and WorkFrame/2 support
- Extensive customization facilities
- Powerful C-like extension language
- Unlimited Undo and Redo
- Powerful search and replace functions
- No limits on file sizes, open files or line lengths

Limited time offer! \$149.00
Order now through Indelible Blue
1-800-776-8284
MSRP \$249.00

PREDATOR/2 is a trademark of
Compuware Corporation. All
other company or product
names are trademarks of their
respective owners.
© 1995 Compuware
Corporation.

COMPUWARE
Sophisticated Software Used Every Day

**Available
and Ready
for OS/2**

Circle Reader Service Number 17

A few words from our competitors:

Prtablty.

Prtbilty.

Portblty.

Portabit.

No matter how they misspell it, it's still not portability. At Zinc, we understand that porting the last 20% of your code takes 80% of your time.

Only Zinc offers complete portability.

With Zinc® Application Framework™ you can develop on the platform you prefer. And since Zinc is the only one that delivers 100% portability, you'll have your application on other platforms as fast as you can recompile. It's part of what makes Zinc the most productive—and affordable—tool a programmer can use.

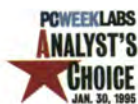
Productivity that leads to opportunity.

Portability is just one road you'll find a little easier. Zinc zips through tedious tasks with C++ object orientation and a unique visual development tool. Globalizing your application is as easy as translating the text. And Zinc is the only product that supplies 100% of the source code.

It all adds up to productivity. Which means more profitability. Which is a concept we probably don't have to spell out for you. For free information and demonstration software, call toll-free:

800 638 8665

Outside the U.S. call: +1 801 785 8900. In Europe call: +44 (0)181 855 9918. In Asia: +81 (052) 733 4301. Contact Zinc electronically at info@zinc.com or GO ZINC on CompuServe.



INFOWORLD
February 6, 1995

"Developers seeking easy delivery of GUI applications on DOS, Windows, OS/2, Macintosh and Unix platforms, or pursuing international markets will find Zinc their best option by far."

"Zinc came closest of all the products we tested to our ideal of portability: just copy the code to the target machine, then recompile and relink the application...In short, Zinc did a great job."

z i n c

NO LIMITS

The Suite Way to Build Portable Applications

Is building applications your job? Then life just got easier thanks to the zApp® Developer's Suite. The zApp Developer's Suite is a set of highly integrated C++ development tools designed to help you transform the blueprints in your mind into commercial quality applications — quickly and easily. And best of all, applications built using the zApp Developer's Suite are portable to fourteen different platforms!

The zApp Developer's Suite consists of zApp, the award-winning portable C++ application framework; zApp Factory™, a fully visual application designer and code generator; and the zApp Interface Pack, a collection of high-level visual objects for the zApp environment. All of these tools are highly integrated to provide maximum ease of use and flexibility.

Rapid Application Development.

Introducing an exciting new visual technology that lets you drag and drop a wide assortment of objects like toolbars,



tables, and 3D dialogs; define their characteristics; and build interfaces of any complexity; all in one powerful but easy to use environment. With the click of a button, you can engage a powerful test mode which lets you interact with your application, seeing it exactly like your end user will see it: letting you fill in dialogs, pull down menus, etc. When you are pleased with the look and feel of your application, fully commented C++ source code is only a mouse click away, thanks to the zApp Developer's Suite's code generation capabilities.

Object-oriented Power.

The best news is that this development environment sits on top of zApp, the industry leading C++ application framework, and the zApp Interface Pack, so you have all of their power at your disposal — toolbars, table objects, advanced graphics — in all, over 300 object classes of power just waiting to be tomorrow's best-selling application.

Portability and More.

When you're done building your application, *then* you can decide what platforms you want to support! Applications built using the zApp Developer's Suite are single-source portable to fourteen different platforms. By simply recompiling, your application will run natively on Windows, Win32 (including Windows 95), OS/2®, and the leading X/Motif platforms.

Free Demo.

Sound impossible? Well, if seeing is believing for you, call **1-800-346-6275**, ask for our free demo disk, and get a glimpse of what the future has to offer.



INMARK

2065 Landings Drive, Mountain View, CA 94043
T: 800-346-6275 or 415-691-9000 F: 415-691-9099

6362-0001-30



zApp and Inmark are registered trademarks of Inmark Development Corporation. zApp Factory is a trademark of Inmark Development Corporation. OS/2 is a registered trademark of IBM. All other trademarks are the property of their respective owners. BBS: 415-691-9990 • Internet: info@inmark.com
CompuServe: GO INMARK
World Wide Web: <http://www.inmark.com/>